



Universidade de Aveiro

Ano 2022

**Diogo Miguel
Azevedo**

**Desenvolvimento da solução de Customer Facing do Grupo
Rangel**

Universidade de Aveiro

Ano 2022

**Diogo Miguel
Azevedo**

**Desenvolvimento da solução de Customer Facing do Grupo
Rangel**

Relatório de estágio apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Informática, realizada sob a orientação científica do Doutor Osvaldo Pacheco, Professor Auxiliar do Departamento de Engenharia Electrónica, Telecomunicações e Informática da Universidade de Aveiro.

o júri

presidente

Prof. Doutor Joaquim Arnaldo Carvalho Martins
Professor Catedrático, Universidade de Aveiro

vogais

Prof. Doutor Jorge Augusto Fernandes Ferreira
Professor associado da Universidade de Aveiro

Prof. Doutor Osvaldo Manuel da Rocha Pacheco
Professor Auxiliar, Universidade de Aveiro

agradecimentos

Gostaria de começar por agradecer á empresa Rangel Logistics Solutions, pela oportunidade do estágio e pelas condições que me proporcionou na realização do mesmo.

Um agradecimento ao meu coordenador de estágio, Professor Osvaldo Pacheco, assim como a todo o corpo docente do Departamento de Eletrónica, Telecomunicações e Informática (DETI) pela formação académica que me forneceram.

Agradeço á minha irmã e aos meus pais por me terem sempre acompanhado neste jornada e me darem sempre a força e confiança para nunca desistir. A todos os meus amigos que ao longo dos anos me acompanharam.

palavras-chave

frontend, backend, track and trace, stored procedures.

resumo

Foi escolhido pelo autor deste trabalho participar num estágio de final de curso de forma a completar o seu Mestrado em Engenharia Informática na Universidade de Aveiro.

Este relatório de estágio tem como objetivo explicar o trabalho desenvolvido na vertente tecnológica na empresa Rangel Logistics Solutions. O autor integrou a DSI (Direção de Serviços e Informática) e a equipa que desenvolve o novo Web-Portal da Rangel.

Neste relatório pretende-se ainda apresentar as principais contribuições do autor para o Web-Portal.

O objetivo principal associado à realização deste estágio foi aprofundar os conhecimentos na vertente Web da informática e ganhar experiência de trabalho num ambiente empresarial.

As tecnologias e metodologias usadas neste estágio foram uma mistura de tecnologias e metodologias já conhecidas com outras novas, quer ao nível *frontend* quer ao nível *backend*.

Globalmente avaliamos este estágio como uma experiência muito positiva, permitindo a melhoria dos conhecimentos nas áreas pretendidas e ficando com melhor conhecimento de como funcionam as dinâmicas de trabalho de uma equipa e de uma empresa na área da informática.

keywords

frontend, backend, track and trace, stored procedures.

abstract

The author of this work chose to participate in an end-of-course internship in order to complete his Masters in Computer Engineering at the University of Aveiro. This internship report aims to explain the work developed in the technological area at the company Rangel Logistics Solutions. The author was part of the DSI (Services and IT Department) and the team that develops the new Rangel Web-Portal.

This report also intends to present what the main contributions of the author to the Web-Portal.

The main objective associated with this internship was to deepen the knowledge in the Web aspect of computing and gain work experience in a business environment. The technologies and methodologies used in this internship were a mixture of technologies and methodologies already known with new ones, both at the frontend and backend level.

Overall, we evaluate this internship as a very positive experience, allowing for the improvement of knowledge in the desired areas and getting a better understanding of how the workflow of a team and a company in the IT area works.

Índice

1. Introdução	1
1.1. Enquadramento e Contextualização	1
1.2. Caracterização do Trabalho	2
1.2.1. História do Portal	3
1.3. Estrutura do Relatório	5
2. Revisão da literatura.....	6
2.1. História dos Web-Portals.....	6
2.2. Portais no âmbito da logística	12
2.3. Introdução	12
2.3.1. Designing a Regional E-Logistics Portal	12
2.3.2. Graph Models and GeoData Based Web Portal in Cargo Transportation	14
3. Descrição das tarefas realizadas.....	19
3.1. Requerimentos de Sistema.....	19
3.1.1. Atores	19
3.1.2. Design de software	21
3.2. Arquitetura do Portal.....	21
3.3. Tecnologias usadas no desenvolvimento do Portal	23
3.3.1. IDE's	23
3.3.2. Linguagens de Programação.....	24
3.3.3. Softwares	26
3.4. Funcionamento do Portal	28
3.4.1. Login e menu principal	28
3.4.2. Histórico de Envios	29
3.4.3. Criar envio.....	32
3.4.4. Libertação de Lote	34
3.4.5. Reclamações	34
3.4.6. Entrada Armazém	35
3.4.7. Registo de Saídas	36
3.4.8. Consulta Stock	36

3.4.9.	Movimentos Armazém	37
3.4.10.	Produtos	37
3.4.11.	Notificações	38
3.4.12.	Gestão.....	38
3.4.13.	Livro de Endereços.....	39
3.4.14.	Embalagens.....	39
3.4.15.	Criar Embalagem.....	39
3.5.	Contribuições no âmbito do estágio.....	40
3.5.1.	Primeiro Projeto	40
3.5.2.	Projetos intermediários e resolução de bugs	41
3.5.3.	Projeto Alcotrans	43
4.	Conclusão e trabalhos futuros.....	49
5.	Bibliografia.....	50

Índice de Figuras

Figura 1 - Rangel Logistics Solutions [3]	1
Figura 2 - Portal antigo da Yahoo (dos primeiros portais a existir) [5].....	7
Figura 3 - Usos de Portais [6].....	8
Figura 4 - Diferentes funcionalidades de um Web-Portal [8].....	9
Figura 5 - Engenharia básica de um web-portal [9].....	10
Figura 6 - Exemplo de uma arquitetura de um web-portal [referência própria]	11
Figura 7 - Colaboração de trabalhos [11]	15
Figura 8 - Alternativa 1 [11]	16
Figura 9 - Alternativa 2 [11]	16
Figura 10 - Uso de um Web Service [11]	17
Figura 11 - Use Case [referência própria].....	20
Figura 12 - Camadas da arquitetura do Portal [referência própria].....	22
Figura 13 - Menu Login [25].....	28
Figura 14 - Menu Principal [25]	28
Figura 15 - Opções Avançadas [25]	29
Figura 16 - Pesquisa em calendário [25].....	29
Figura 17 - Pesquisa Avançada [25].....	30
Figura 18 - Ações avançadas (1) [25]	30
Figura 19 - Ações avançadas (2) [25]	31
Figura 20 - Informação da Viagem [25]	31
Figura 21 - Temperatura Ambiente de Veículo [25].....	31
Figura 22 - Criação de Envio (Expedidor) [25]	32
Figura 23 - Criação de Envio (Dados Envio) [25].....	33
Figura 24 - Criação de Envio (Volumes) [25].....	33
Figura 25 - Criação de Envio (Volumes) [25].....	33
Figura 26 - Libertação de Lote [25].....	34
Figura 27 - Opções Avançadas (3) [25]	34
Figura 28 – Reclamações [25].....	34
Figura 29 - Entradas Armazém [25]	35
Figura 30 - Pesquisa Rápida [25].....	35
Figura 31 - Detalhe Documento [25]	35
Figura 32 - Registo de Saídas [25].....	36
Figura 33 - Consulta Stock [25].....	36
Figura 34 - Movimentos Detalhe [25].....	37
Figura 35 – Produtos [25]	37
Figura 36 - Criação de Notificação (1) [25]	38
Figura 37 - Criação de Notificação (2) [25]	38
Figura 38 - Painel Gestão [25].....	38
Figura 39 - Livro de Endereços [25]	39
Figura 40 – Embalagens [25]	39
Figura 41 - Processo GCOM [referência própria]	41

Figura 42 - Envio de credenciais [25].....	42
Figura 43 - Roles de utilizador (antigo) [25]	42
Figura 44 - Roles de utilizador (novo) [25]	43
Figura 45 - Mapping dos dados [referência própria].....	44
Figura 46 - Render e gerar XML [referência própria]	45
Figura 47 - Escrita e upload do ficheiro [referência própria]	45
Figura 48 - Obter ficheiros de Track and Trace [referência própria].....	46
Figura 49 - Ciclo de parse do ficheiro [referência própria].....	46
Figura 50 - Criação/Update envios e eventos [referência própria].....	47
Figura 51 - Obter ficheiros de Proof-Of-Delivery [referência própria].....	47
Figura 52 - Obter ler e fazer mapeamento do Proof-Of-Delivery [referência própria]	48
Figura 53 - Criar string Json, chamada RestAPI e eliminar ficheiro [referência própria]	48

Lista de acrónimos e de estrangeirismos

DSI = Direção de Serviços e Informática
API = Application Programming Interface
HTTP = Hypertext Transfer Protocol
WBF = Web Based Framework
SOA = Arquitetura orientada a serviços
IDE = Integrated Development Environment
DOM = Document Object Model
ICOL = Sistema de gestão de armazéns
GOM = Sistema de gestão operacional de distribuição expresso.
RDL = Rede logística
RT = Rede Transitária

1. Introdução

A unidade curricular dissertação/estágio (código 49997) no âmbito do Mestrado de Engenharia Informática da Universidade de Aveiro foi concretizada no estágio que decorreu na Rangel Logistics Solutions.

Neste primeiro capítulo é apresentada a empresa onde decorreu o estágio, são descritos os principais processos que esta empresa apresenta no mundo da logística e um enquadramento dos objetivos do estágio. Por último é apresentada a estrutura deste relatório, assim como uma breve descrição de cada ponto.

1.1. Enquadramento e Contextualização

Este estágio foi realizado na empresa Rangel Logistics Solutions, integrando o Departamento de Sistemas de Informação no grupo de Desenvolvimento do Portal.

A Rangel Logistics Solutions é uma empresa de logística portuguesa, com sede na Maia e fundada em 1980 por Eduardo Rangel [1]. Esta empresa é reconhecida como um parceiro logístico global oferecendo ao mercado uma solução *One Stop Shop* (um ambiente virtual ou físico, em que o consumidor pode realizar compras de diferentes itens num só local). Esta empresa move mercadorias entre mais de 220 países e territórios seja por terra, mar e via aérea [2] (Figura 1).

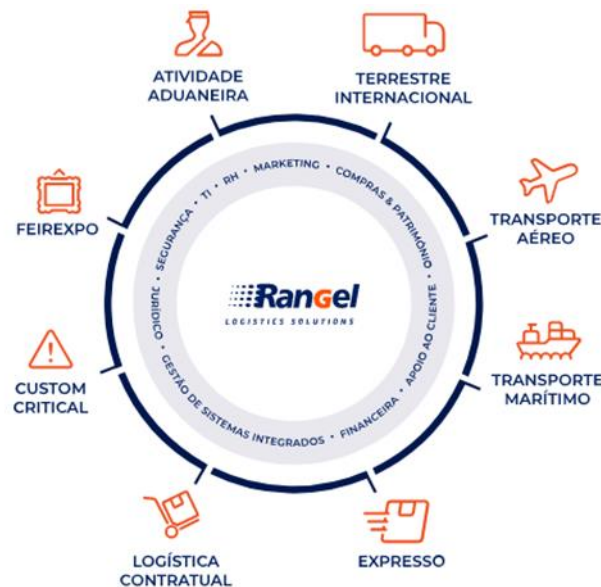


Figura 1 - Rangel Logistics Solutions [3]

1.2. Caracterização do Trabalho

Este estágio decorreu na Direção da vertente tecnológica da empresa, DSI (Direção de Serviços e Informática) e integrado a equipa que trabalha no desenvolvimento do novo *Web-Portal* da Rangel, um projeto já em curso. O autor participou no desenvolvimento de um conjunto de funcionalidades a serem lançadas para o mercado.

Com este estágio foram aprendidas um novo conjunto de tecnologias e também o aprofundamento de algumas previamente conhecidas, tais como *HTML, CSS, JavaScript, TypeScript, Angular, C#, .Net Core e SQL Server*.

O Portal Único do Cliente Rangel, plataforma cuja marca comercial é *MyRangel*, é uma das mais estratégicas e importantes plataformas de *customer facing* (interação ou comunicação diretamente com os clientes) do grupo. Permite aos seus clientes acesso em tempo real a tudo o que está a ocorrer na sua cadeia logística desde informação de *Track and Trace*, acesso a documentos, monitorização e localização de frota (através da integração com sistemas de monitorização de temperatura e GPS), processos de *inbound* (entradas de armazém), de *outbound* (saídas de armazém) e um conjunto de processos relacionados com a Qualidade.

O Portal da Rangel é uma evolução do portal existente tendo várias novas funcionalidades, sendo algumas delas:

- Melhoria significativa da performance;
- Adaptação aos vários suportes (desktop, mobile e tablet);
- Automatização da impressão de etiquetas e manifestos;
- Simplificação da criação de envios (através do pré-preenchimento de campos repetidos);
- Portal Multilíngue;
- Possibilidade de personalização de medidas de embalagens nos envios;
- Possibilidade de personalização do perfil, com o logotipo do Cliente;
- Melhoramento do *Track and Trace*;

1.2.1. História do Portal

O primeiro portal a ser desenvolvido foi o *Rangel Portal*, este portal ainda continua em funcionamento. O *MyRangel* (portal cujo desenvolvimento o autor participou no âmbito deste estágio) baseou a sua imagem na imagem do *Rangel Portal*.

Numa cadeia de distribuição existem vários estados, como por exemplo: Entregue, Não entregue, *Pickup*, Documento Validado, etc. Foi registada a necessidade de comunicar aos clientes da logística quais seriam os estados dos envios, em função da cadeia de distribuição.

Com isto também surgiu a necessidade de haver um controlo de temperaturas (do veículo), por causa do material farmacêutico e um relatório de viagem onde se encontram as informações de todos os envios, dos volumes e os estados pelo qual eles passam.

Normalmente os envios começam a partir de um armazém da Rangel em que os trabalhadores do armazém fazem a gestão dos produtos. Estes produtos são inseridos numa ordem de saída através de um menu do portal. Com o desenvolvimento do *Rangel Portal* foram introduzidas novas funções como o controlo de lote e o módulo de reclamações. O *Rangel Portal* foi fundamental para começar com a introdução da vertente informática/web para o negócio da logística da Rangel, principalmente no ramo da logística de distribuição farmacêutica.

No âmbito da atividade da Rangel surgiu a necessidade de recolha de encomendas nas instalações próprias de uma farmacêutica ou armazém farmacêutico e a integração diretamente com o cliente, em vez de utilizar um armazém da Rangel e os seus recursos para fazer o controlo dos serviços. Passa a ser a empresa farmacêutica a realizar este controlo e a distribuição faz-se a partir dessa empresa. Uma grande vantagem desta alteração do método de trabalho é que uma empresa pode estar situada longe das instalações da Rangel e, a partir do momento que um cliente cria um envio, cria também uma etiqueta que solicita um pedido de recolha de um veículo da Rangel. Para suporte a estas novas alterações foi desenvolvido um novo website, o *Shipping*.

O *Shipping* é onde é permitido fazer o registo de um envio. A criação do envio é feita com as características particulares que são necessárias para levar a cabo o transporte, como o endereço do destino, uma referência alusiva ao transporte, autoridade tributaria, número de volumes, etc...

Com a introdução deste portal, deixou de ser exclusivo o uso dos armazéns da Rangel e assim passou a ser possível expandir para clientes que têm armazéns próprios e permitir que eles controlem todo o processo de expedição a partir das suas instalações. Um exemplo de um caso de utilização do *Shipping*: o cliente cria uma etiqueta solicitando a recolha

através de uma viatura da Rangel, com temperatura controlada, para fazer o envio para o cliente final. A partir do momento em que a viatura da Rangel recolhe o envio do armazém é possível fazer o *Track and Trace* para acompanhar todos os estados a partir do *Rangel Portal*.

A coexistência de dois *websites* é obviamente inconveniente para os clientes e para a própria empresa. Avançou-se então para o desenvolvimento do *MyRangel* e a integração do *Shipping*, tendo uma melhor estrutura e apresentação.

Aproveitando a estrutura já montada do *Rangel Portal*, foram alargadas as suas funções e métodos para outros serviços complementares da Rangel, como o serviço expresso, conhecido como *Correos Express*, o *Despachante*, e integrou-se alguns clientes, como os transportes rodoviários internacionais e transportes marítimos. Também foi possível o uso *Track and Trace* para o controlo do estado da encomenda durante o percurso na cadeia de distribuição.

Correos Express

Identificando a necessidade da integração do módulo de *Shipping* para a distribuição nacional identificou-se uma oportunidade: “já que na parte da logística temos infraestruturas para fazermos a distribuição em 24 horas, porque não criar um serviço em que pudéssemos fazer o mesmo, mas ‘*Bussiness-To-Consumer*’?”. Com isto foi dada a oportunidade a alguns clientes da logística para poderem entrar na rede da distribuição da Rangel, sem acesso a algumas funcionalidades, como o registo de temperaturas (visto que não era necessário). Assim, foi adaptado o *Shipping* para fazer esta função expresso, em parceria com a empresa Correos, dando assim origem ao Correos Express, previamente conhecido como Rangel Express. Este portal tem quase as mesmas funções do portal *Shipping*, exceto o *Track and Trace* e o registo de temperaturas, porque desde o momento em que a etiqueta é criada o envio é registado como encerrado.

1.3. Estrutura do Relatório

O relatório está dividido em 4 capítulos, sendo estes:

- Capítulo 1 (Introdução) – Neste capítulo é feita uma introdução ao estágio, à empresa onde decorreu o estágio e ao Portal da Rangel.
- Capítulo 2 (Revisão da Literatura) – Neste capítulo é feita uma revisão da literatura onde se apresenta a evolução dos *web-portals* e onde se apresenta alguns artigos estudados;
- Capítulo 3 (Trabalho) – Neste capítulo descrevo as tecnologias usadas no desenvolvimento do portal e apresento as contribuições do autor a este projeto;
- Capítulo 4 (Conclusão e trabalhos futuros) – Neste capítulo apresento algumas reflexões sobre o estágio, descrevendo o que de mais importante foi desenvolvido e as lições aprendidas.

2. Revisão da literatura

Neste capítulo apresenta-se uma revisão do estado de arte sobre portais no contexto da área da logística.

Foram analisados artigos em que se analisou métodos diferentes de implementação de portais para a logística, descrevendo-se as vantagens e desvantagens respetivas.

2.1. História dos Web-Portals

A palavra Portal veio da palavra latina *porta*, que é traduzida como porta. Qualquer coisa que efetue a função como uma porta de entrada para qualquer outra coisa é considerado um portal.

Um *web-portal* é uma plataforma baseada na web que reúne informações de fontes diferenciadas em uma única interface do utilizador e fornece aos mesmos as informações mais relevantes com base no seu contexto. Com o tempo, os *web-portals* evoluíram para plataformas de portal que suportam a experiência digital do cliente.

Os três pontos fortes dos portais são a integração, a consistência e a personalização.

- Os recursos de integração permitem a ligação entre dados e sistemas de dados do cliente no *backend*;
- É necessária consistência através dos pontos de contacto digitais, fornecendo a estes sistemas conectados aparência semelhantes no *frontend*;
- Também é necessária uma personalização através de uma combinação de dados comportamentais e biográficos que concedem às empresas a capacidade de contextualizar experiências. Normalmente isto é feito com o *login*.

Pode haver uma confusão entre a distinção de um *website* e um *web-portal*, apesar de haver algumas semelhanças, também existem algumas diferenças:

- Num *web-portal* não é normal aceder por um *URL* definido, o utilizador tem de seguir certos passos para ter acesso, sendo preciso ter acesso ao mesmo para utilizar as suas ferramentas;
- O *web-portal* é centrado na experiência do utilizador, enquanto um *website* centra mais na imagem da empresa;
- No *website* existem atualizações de forma muito mais regulares quando comparado com *web-portal*.

O servidor de um portal age como *gateway* (porta de entrada) para a rede de uma empresa.

Hoje em dia as empresas exigem uma coordenação de várias fontes de dados, processos, utilizadores e o modo como partilham as informações entre eles. Para conseguir isso de forma mais eficaz e mais bem estruturada, é necessário algum tipo de *business intelligence* (inteligência de negócio) ou aplicativos para gerir o conhecimento e para sintetizar/compreender os dados. Este nível de exigência é um dos motivos que levou ao desenvolvimento de portais.

Uma das vantagens mais destacada dos *web-portals* é a necessidade de autorização para poder aceder às informações pelos utilizadores [4], fazendo com que toda a informação das empresas e dos utilizadores sejam mais seguras. Também faz com que informação e processos sejam mais bem estruturado e assim mais eficientes.



Figura 2 - Portal antigo da Yahoo (dos primeiros portais a existir) [5]

Assim, há muitos benefícios que os portais oferecem aos negócios e às empresas. Melhor tomada de decisão, melhor comunicação e maior produtividade ou até mesmo redução de custos. Antes da introdução da tecnologia, num ambiente empresarial era normal as tarefas serem algo complexas e de certa forma caras. Hoje em dia continuam a ser complexas, mas agora temos as tecnologias para conseguir acompanhar este nível de complexidade. Com estas novas tecnologias, houve uma certa motivação às empresas para investir em *web-portals*, de forma a fazer uma melhor gestão de informação e mais eficiente organização.

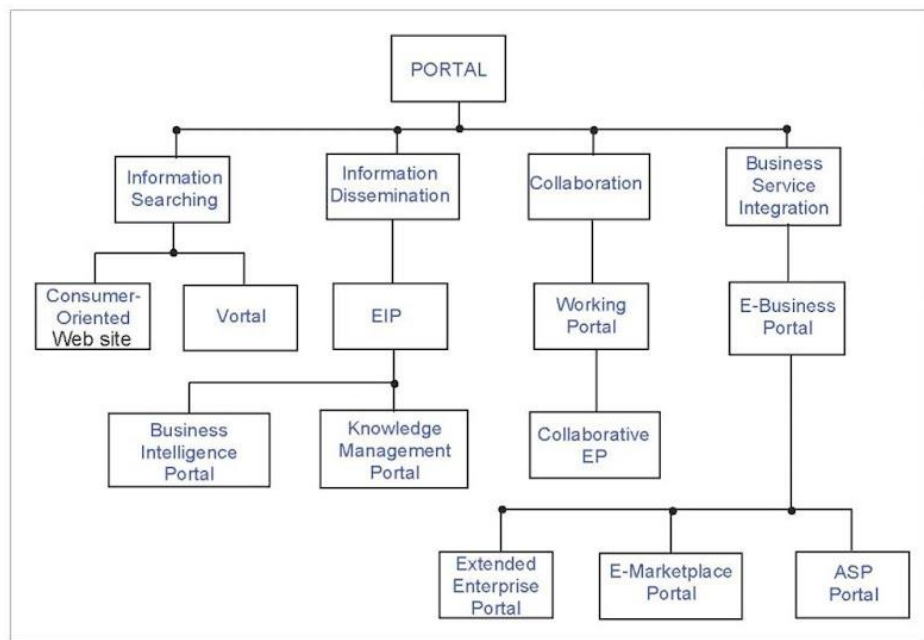


Figura 3 - Usos de Portais [6]

Um portal consegue fornecer um único ponto de acesso para funcionários, parceiros e clientes a vários tipos de informações, dando assim uma importante contribuição para permitir a gestão e o conhecimento das empresas. Os *web-portals* também fornecem inteligência de negócio e ferramentas colaborativas, potenciando vantagens competitivas significativas para quem as usam.

Os portais podem ser classificados como horizontais ou verticais [7].

Um portal horizontal é usado como plataforma para várias empresas de um setor económico ou para o mesmo tipo de fabricantes ou distribuidores. Um portal vertical é um ponto de entrada especializado para um tipo de mercado ou setor específico.

Os portais podem ser agrupados em vários grupos, sendo estes 3 os mais importantes [7]:

1. Portais públicos (Public portals);
2. Portais pessoais (Personal portals);
3. Portais corporativos (Corporate portals).

Os **portais públicos** seguem um modelo em que fornece serviços como imagens, e-mails, fóruns, etc. Este portais tendem a atingir uma demográfica maior e uma maior audiência de utilizadores online.

Exemplos de portais públicos são o Google e o Yahoo.

Os **portais pessoais** podem ser um website que normalmente é feito com o intuito de ser usada para aplicações distribuídas, isto é, irão ser utilizados um número de *middleware* (software usado por diferentes aplicações para se comunicarem entre si) e *hardware* (conjunto dos componentes que

compõem a parte material de um computador) diferentes de forma a fornecer serviços de diferentes fontes. Os conteúdos destes portais podem ser apresentados em diferentes plataformas como *laptops*, *PC's*, *tablets*, telemóveis (*smartphones*), etc...

Os **portais corporativos** são mais comuns a partir dos anos 90. Com o “crescimento” da internet, o seu nível de complexidade vai aumentando, e com isto novos desafios vão surgindo e um maior nível de dificuldade no seu desenvolvimento. Assim, os recursos das empresas tornaram-se insuficientes para lidar com estes desafios. Os portais corporativos ajudaram as empresas a conseguirem gerir melhor a qualidade dos seus dados e informações. Com o surgimento de novos padrões (*JSR168* e *JSR*) permitirão que os *developers*, administradores e consumidores de portais integrassem portais e *portlets* baseados em padrões disponibilizando uma



Figura 4 - Diferentes funcionalidades de um Web-Portal [8]

variedade de soluções de fornecedores. O padrão *JRS* permite a interoperabilidade entre *portlets* de plataformas de diferentes portais. Os portais corporativos também oferecem oportunidades de atendimento automático para clientes e funcionários.

Outras soluções destes portais podem incluir a melhor gestão de fluxo de trabalho (*workflow*), uma melhor colaboração entre grupos de trabalho ou filiais e publicação de conteúdos. Estas funcionalidades são as mais importantes no contexto deste trabalho, visto que estas são mais exploradas e foram implementadas a prática.

Também existem outros tipos de portais, como por exemplo portais de governos, portais de procura (motores de busca), portais culturais (bibliotecas) e portais de propriedades.

Na perspetiva da engenharia, o conceito principal é apresentar ao utilizador uma única página web em que se possa reunir ou agregar conteúdos de vários outros sistemas ou servidores em que estes servidores estão conectados a servidores de uma base de dados e podem fazer parte de um ambiente de servidor em cluster.

As tecnologias mais recentes do lado do cliente (*client-side*) são estruturas e bibliotecas *JavaScript/TypeScript* que contam com funcionalidades da Web mais recentes, como *WebSockets* e retornos de chamada assíncronos usando *XMLHttpRequests*.

Normalmente numa arquitetura tem-se dois tipos de programação, *backend* e *frontend*. O *backend* refere-se à gestão dos dados, criação de serviços e implementação de conexões com bases de dados e à transposição dos dados para o *frontend*. Para o *backend* utiliza-se um conjunto de ferramentas, e linguagens de programação como *C*, *C++*, *C#*, *Java*, entre outras e também *SQL* para a programação das bases de dados.

No *frontend* desenvolve-se a camada de apresentação para os utilizadores. Normalmente usa-se *HTML*, *CSS*, *JavaScript*, *TypeScript*, *Angular.js*, *React.js*, etc...

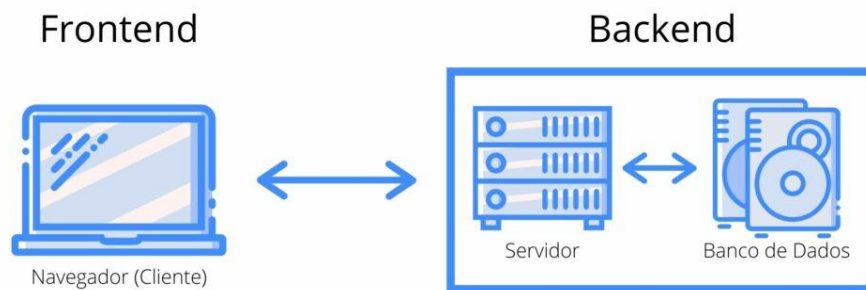


Figura 5 - Engenharia básica de um web-portal [9]

A arquitetura de um portal pode ser representada por certas camadas e por um fluxo. No exemplo seguinte é apresentado algo similar ao que foi utilizado no estágio (Figura 6):

- *Database*: Estrutura onde os dados serão guardados, tanto registos de compras, credenciais de utilizadores, etc...;
- *Entity Framework*: Transforma dados em objetos.
- *WebApi* interna: É uma *API (Application Programming Interface)* que pode ser acedida através de protocolos (*HTTP* por exemplo);
- *Repositório*: Servirá como uma camada abstrata entre a lógica da nossa aplicação e o acesso aos dados;
- *Domain*: Irá servir para a implementação de classes, modelos e o *mapping* da nossa base de dados;

- *Controller*: Módulo onde estão os controladores que disponibilizam os métodos públicos;
- *WebApi* externa: similar à *API* interna; serve como uma camada de proteção, tornando os dados mais protegidos e menos comprometidos a certas infrações;
- *Frontend*: Parte visível pelo utilizador;

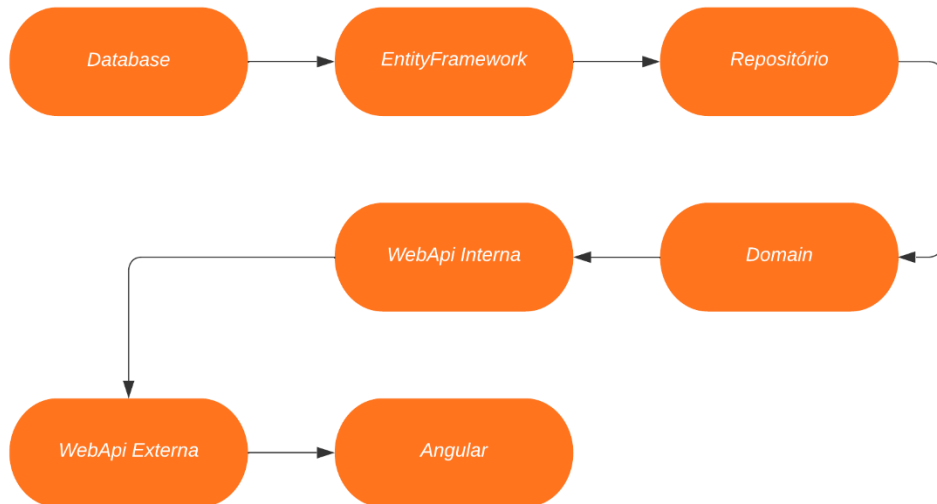


Figura 6 - Exemplo de uma arquitetura de um web-portal [referência própria]

Ao longo dos anos houve várias tecnologias usadas no desenvolvimento dos vários módulos presentes na arquitetura de um *web-portal*. Hoje em dia as mais usadas no *frontend* são o *JavaScript* juntamente com *HTML* e *CSS*, isto no contexto de *web programming* [10].

Outras linguagens de programação conhecidas incluem o *Ruby*, *Python*, *TypeScript*, *Java*, *PHP*, *SQL*, entre outras.

Antigamente as aplicações usadas eram monolíticas. Elas eram todas escritas apenas em uma unidade de código e todos os componentes partilhavam a mesma memória e os mesmos recursos [10].

Hoje em dia as arquiteturas usam *API's* (*Application Programming Interface*) e *microserviços*, fazendo com que sejam levemente acopladas. Algumas das arquiteturas mais usadas ao longo dos anos são:

- Arquiteturas Monolíticas;
- Arquiteturas de *Microserviços*;
- Arquiteturas Orientadas por eventos;
- Arquiteturas Orientadas a serviços.

2.2. Portais no âmbito da logística

No âmbito da logística, a existência de portais que facilitem a interação com o utilizador são de extrema importância. Este tipo de abordagem tem vindo a crescer e a ser adotada por facilitar a forma amigável como disponibiliza o acesso aos seus serviços sem necessidade de aplicações específicas aos seus utilizadores.

2.3. Introdução

Foram estudados alguns artigos no âmbito de portais na área da logística, com o objetivo de fazer um estudo sobre trabalhos prévios na mesma área, avaliando diferentes métodos de criação de *web-portals* e respetivas vantagens e desvantagens.

Infelizmente os artigos que resultaram da pesquisa realizada pouco continham sobre o que era procurado, mesmo assim, ainda foi possível retirar alguma informação, visto que mostrava aspetos similares ao que foi usado no estágio.

2.3.1. Designing a Regional E-Logistics Portal

Este artigo, da autoria de Adrian Collins, George Ditsa, Anindya Ghose, Peter Hyland e Sim Lau [2005] tem como propósito desenvolver um portal para *regional e-logistics* (logística online).

E-logistics é definido como a gestão de todos os fluxos físicos de uma organização que vende mercadorias em uma plataforma online. Por suas muitas particularidades, a *e-logistic* representa uma questão crucial para os *e-comerciantes* e requer uma implementação de ações e processos específicos para que um *e-comerciante* consiga beneficiar de uma ótima gestão de fluxo (*workflow*).

O propósito deste projeto era tornar as tecnologias de otimização baseada em restrições e negociação automatizada acessíveis a empresas pequenas e médias, dentro de um portal que também atende às suas necessidades de informação.

Na visão dos autores os três aspetos essenciais de um portal de *e-logistics* são:

- Informação (visibilidade e troca);
- Otimização;
- Negociação.

Os maiores desafios neste projeto foram:

- Armazenamento de dados: a troca de informações exige que os horários de produção para fornecedores, os horários de produção para clientes e os horários de entregas para os fornecedores de logística sejam armazenados em algo como uma base de dados.
- Visualização de dados: a visualização da informação em toda a corrente de suplementos é exigida não apenas por máquinas automatizadas de suporte a certas decisões, mas também pelos utilizadores humanos. Um desafio importante seria determinar como as decisões de logística podem ser visualizadas de maneira significativa. O *design* escolhido utilizava gráficos de Gantt interativos.
 - Estes gráficos traziam algumas vantagens. Eles abriam caixas de texto quando o cursor era movido sobre partes específicas do gráfico contendo assim certos detalhes relevantes, como os preços dos itens, critérios de qualidade, etc... O gráfico de Gantt interativo permitia aos utilizadores também alterar horários através de uma interface *drag-and-drop*.
- Políticas de acesso a dados: As informações de logísticas relativas a uma determinada empresa só podem ser parcialmente reveladas a outras empresas. Com isto as empresas normalmente exigem que apenas os subconjuntos mínimos necessários para coordenar os horários ou para chegar a resultados de negociação bem-sucedidos sejam revelados a outras partes.

O *design* dependia de um tipo de tecnologia chamado *constraint programming*. Isto é um paradigma de programação que se refere ao uso de restrições na construção de relações entre variáveis. Consiste em especificar, para uma solução, que critérios (restrições) esta tem de cumprir. Este paradigma começou a ser desenvolvido nos anos 70.

Vantagens de usar constraint programming:

1. A *constraint programming* suporta uma linguagem de representação de conhecimento mais robusta que permite que todos os conhecimentos/restrições relevantes ao domínio sejam utilizados no processo de resolução de problemas.
2. A *constraint programming* oferece uma linguagem de modelação mais robusta e flexível para especificar restrições. As técnicas de programação são uma mistura de linear e inteira, mas por outro lado, sobrecarregam o utilizador na modelação de problemas numa linguagem menos flexível, que insiste que todas as restrições sejam expressas como relações lineares.
3. A linguagem flexível de representação do conhecimento suportada pela *constraint programming* pode, em alguns casos, permitir a especificação de restrições que reduzem significativamente o tempo de busca e aceleram drasticamente o processo de resolução.

4. A linguagem de representação do conhecimento permite a interpretação lógica de cada restrição como uma sentença em uma linguagem de primeira ordem. Isso permite a integração da *constraint programming* com uma variedade de outros sistemas formais derivados do campo da inteligência artificial, com benefícios significativos em domínios com informações incertas e mutáveis.

2.3.2. Graph Models and GeoData Based Web Portal in Cargo Transportation

Este artigo, da autoria de Leonids Novickis, Svetlana Vinichenko, Mihails Sotnichoks, Andrejs Lesovskis e Darja Amalitskaya [2015] aborda a criação de um *Web Based Framework (WBF)/ Web Portal* para apoiar os processos de negócios de transporte na área da logística.

Neste artigo os autores afirmam que o transporte intermodal de cargas envolve o uso de dois ou mais modos de transporte em uma rede ligada para o movimento contínuo de mercadorias. Afirmam também, que o transporte de mercadorias que envolve a transferência direta de equipamentos entre modos, sem qualquer movimentação de mercadorias transportadas está ligado a esta rede intermodal. As vantagens do transporte intermodal atraem cada vez mais empresas a integrá-lo nas suas estratégias logísticas. Como resultado, o tráfego intermodal continua a crescer.

Geralmente, as empresas lucram com a logística intermodal nos seguintes aspetos:

1. Transporte porta-a-porta;
2. Redução de custos de transporte;
3. Requisitos de impacto ambiental;
4. Integração facilitada dos processos de produção/consumo por meio de uma ótima gestão da cadeia logística.

Neste artigo também é feita uma análise das diferentes abordagens adotadas por esses sistemas que mostra, que a interoperabilidade baseada na Internet de tecnologias inteligentes é a combinação mais promissora de soluções que podem ser usadas, para criar a próxima geração de estruturas e portais de negócios inteligentes para serviços de logística de transporte.

Funções principais do portal

1. Permitir que os utilizadores organizem um trabalho colaborativo entre representantes de dois grupos-alvo: Transporte e Carga (Imagem 7).

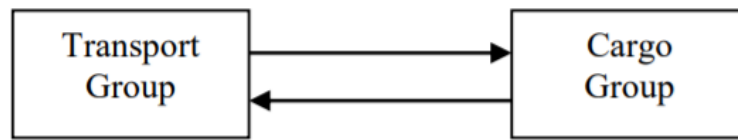


Figura 7 - Colaboração de trabalhos [11]

2. Para o grupo de proprietário de carga:
 - a. Calcular os custos de transporte para entrega da carga desde o local de compra até o ponto de destino de acordo com os *INCOTERMS* 2010;
 - b. Calcular o tempo de trânsito para entrega da carga no ponto de destino;
 - c. Escolher a empresa de transporte que corresponda aos requisitos de um proprietário de carga.
3. Para empresas do grupo de transporte:
 - a. Envolver novos clientes e aumentar a rotatividade de cargas;
 - b. Reduzir a carga de trabalho dos especialistas de uma empresa oferecendo aos potenciais clientes tarifas de transporte que já foram calculadas e guardadas na base de dados do portal, mas não exigidas e utilizadas por alguns motivos;
 - c. Aumentar o lucro de um negócio usando tecnologias avançadas baseadas na Internet;
4. Além disso, ambos os grupos de utilizadores podem receber “Informações úteis” sobre logística de transporte e áreas afins:
 - a. Serviços de linhas de navegação e comboios de blocos;
 - b. Educação e treino;
 - c. Soluções de *IT* e móveis.

Soluções tecnológicas

A integração deste exemplo foi implementada seguindo os princípios da Arquitetura Orientada a Serviços (*Service-Oriented Architecture*). Existem muitos portais e sites independentes de pesquisa na Internet, que oferecem diferentes informações científicas e tecnológicas. *WBF* (*Web Based Framework*) combina todos esses recursos eletrônicos num portal. O principal benefício do *WBF* para os utilizadores é que eles podem ter um único ponto de acesso às informações combinadas.

Do ponto de vista tecnológico, o objetivo principal foi criar a melhor comunicação entre *WBF*, *websites* e portais separados. Em ligação com isso, os serviços da Web são usados como “comunicações-*endpoints*”. Eles podem ser implementados usando diferentes tipos de tecnologias de software: por exemplo, *.NET*, *Java*, *PHP* e outros.

Foi desenvolvida uma *API* com base em *Web Services*. Com a *API*, pode ser feita a transferência de informações entre portais/*websites* e o *WBF*. Para a hospedagem de uma *API* foram investigadas duas alternativas:

Alternativa 1: Uma *API* é integrada ao *WBF*. Neste caso, recursos eletrônicos separados desempenharão o papel de um Cliente (Figura 8).

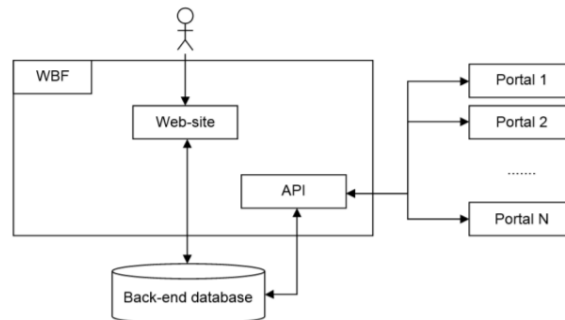


Figura 8 - Alternativa 1 [11]

Alternativa 2: Cada e-recurso (portal web) oferece sua própria *API*, enquanto o *WBF* funciona como um Cliente (Figura 9).

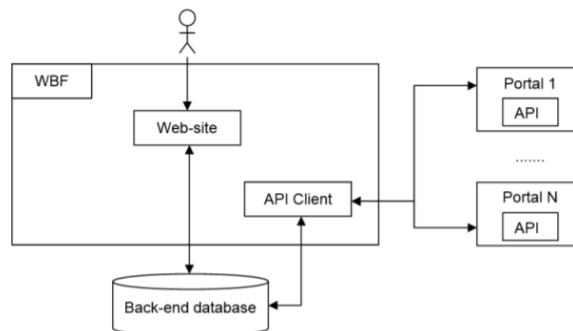


Figura 9 - Alternativa 2 [11]

Ambas as variantes apresentam vantagens e desvantagens, que foram analisadas e discutidas. A 1ª alternativa foi selecionada para implementação do *eINTERASIA WBF*.

Existem vários benefícios da abordagem de integração *SOA* (arquitetura orientada a serviços) e o mais importante deles é tornar-se mais ágil. A arquitetura *SOA* permite que os parceiros de negócios respondam rapidamente a novas condições de negócios e desenvolvam novos recursos. A linguagem *XML* é usada para definir a arquitetura da mensagem e os formatos da mensagem. Um dos principais objetivos do projeto *SOAP* é a simplicidade e extensibilidade. *REST* define um conjunto de princípios de arquitetura pelos quais os *Web Services* podem ser projetados. Os *Web Services* concentram-se num recurso do sistema, incluindo como os estados dos recursos são endereçados e transferidos por *HTTP* por uma ampla gama de clientes escritos em diferentes linguagens. Após analisar tanto *SOAP* quanto *REST*, concluíram que ambas as abordagens funcionam e ambas apresentam vantagens e desvantagens na interface com serviços web.

No caso deles (nível empresarial *WBF*), *SOAP* foi a melhor escolha por vários motivos:

1. Suportar diferentes protocolos e tecnologias, incluindo *WSDL*, *XSDs*, *WS-addressing*;
2. Projetado para lidar com ambientes de computação distribuídos;
3. Mais adequado para serviços intermediários;
4. Tem um grande número de padrões de suporte para segurança, confiabilidade, transações etc.

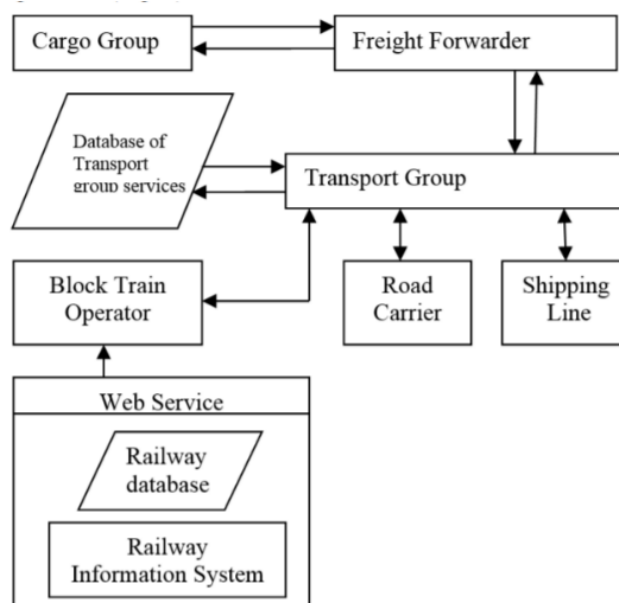


Figura 10 - Uso de um Web Service [11]

Este artigo está mais próximo da abordagem do sistema do portal usado visto

que este usa *SOAP* entre outras tecnologias descritas, podendo ver uma evolução do que poderia ser feito com o *SOAP* já em 2015.

Com o estudo destes 2 artigos (apesar de terem pouca informação), é possível ver uma evolução no *web-portals* na área da logística.

3. Descrição das tarefas realizadas

Este capítulo está dividido em 5 partes. Começo por abordar os requisitos do sistema. Na segunda parte, descrevo a estrutura da arquitetura do Portal Único do Cliente Rangel. Na terceira parte, apresento as tecnologias usadas pelo portal. Na quarta parte, apresento as funcionalidades do portal e na última parte, abordo as várias contribuições do autor.

3.1. Requisitos de Sistema

Nesta seção apresento requisitos do sistema, discuto os atores e apresento diagramas de *use-case* do Portal Único do Cliente Rangel.

3.1.1. Atores

O público-alvo são os **clientes** da Rangel que tencionam usar o portal para fazer encomendas e seguir o processo das mesmas. É esperado dos clientes da Rangel terem um certo conhecimento do uso deste tipo de tecnologias, mas para ajudar, a equipa do portal tem sessões com clientes onde se pretende melhorar a utilização e o esclarecimento de dúvidas. Também está disponível para todos os utilizadores, um manual de uso disponível em quase todas as páginas do portal.

O manual e as formações, são de grande importância visto que em cada mês o portal evolui de diversas maneiras, acrescentando funcionalidades ou apenas corrigindo erros.

Entidades: as entidades representam as empresas que fazem parte da rede de logística. Cada entidade irá ter um conjunto não fixo de funcionalidades, dependendo dos acordos com a Rangel. Um exemplo será que apenas algumas empresas têm acesso a criação de envios, enquanto outras podem criar envios, mas não podem visualizar o histórico de envios.

Clientes: Cada cliente será um utilizador com uma conta própria que irá pertencer a uma certa entidade. Estes utilizadores são quem podem usufruir das funcionalidades que são permitidas à sua entidade.

Administradores/Developers: têm acesso a todas as funcionalidades do portal e a toda a informação armazenada na base de dados.

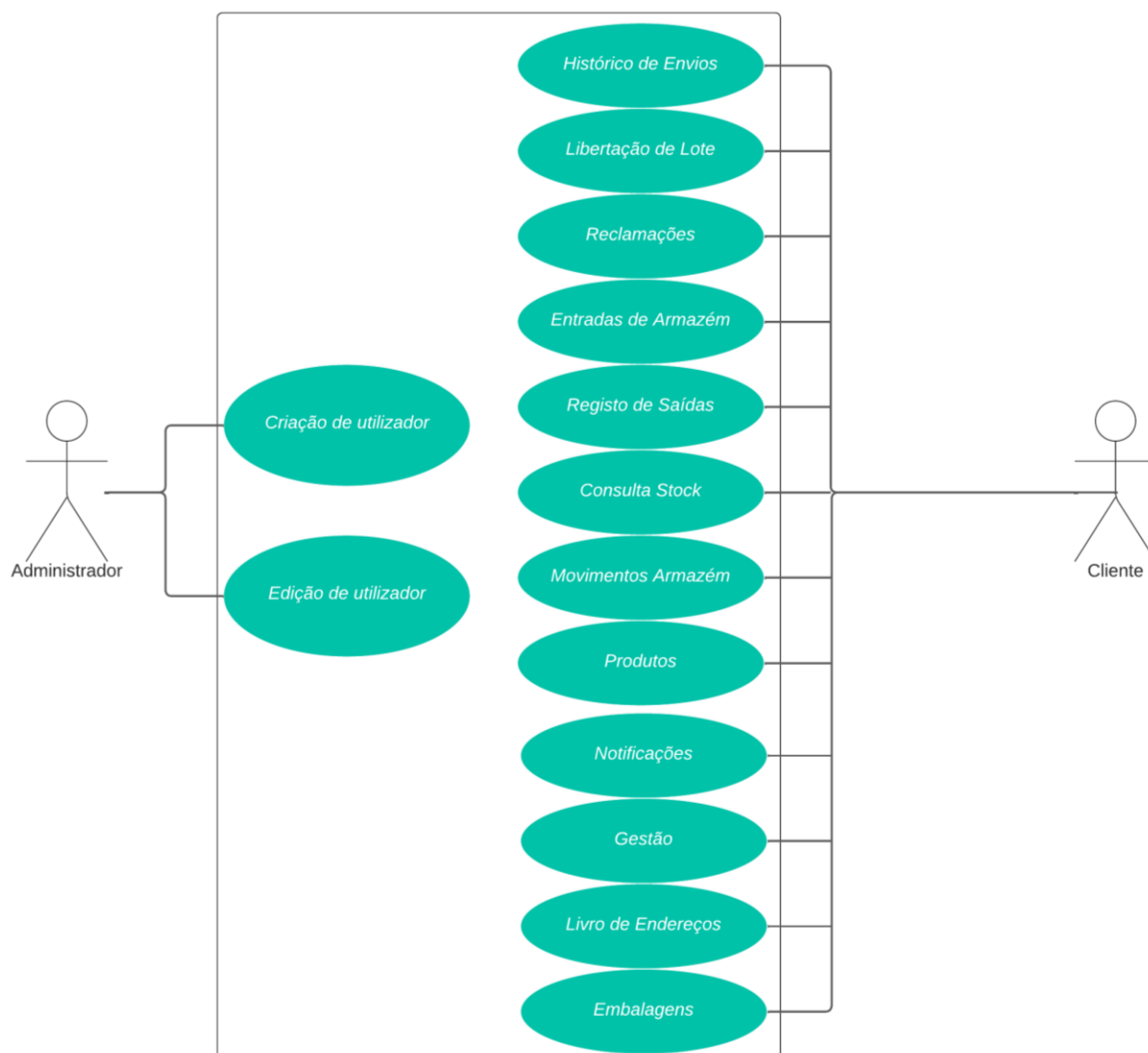


Figura 11 - Use Case [referência própria]

Apesar de o administrador ter acesso a tudo, as funções mais importantes são:

- Criação de utilizador: Criação do utilizador para o cliente, atribuir as credenciais e envio de e-mail com as mesmas e atribuição de permissões e associação a uma entidade.
- Edição de utilizador: Edição dos privilégios que foram atribuídos aquando da criação do utilizador.

As funcionalidades do cliente variam em função do tipo de cliente. Na secção 3.3. (Funcionamento do Portal) o autor explica com mais detalhe as funcionalidades mais importantes.

3.1.2. Design de software

O *design* de software é o processo em que uma especificação de um software é criado com o objetivo de atingir certos objetivos (normalmente resolver problemas), usando um conjunto de componentes sujeitos a restrições [12]. Estes *designs* de software tendem a melhorar o software em certos pontos como a sua segurança, usabilidade, portabilidade entre outros. O *software* do Portal usa o *Domain-Driven Design (DDD)*.

Domain-Driven Design é uma abordagem para o desenvolvimento de *software* que é centrada no desenvolvimento da programação de modelação de um domínio que possui uma compreensão dos processos e regras desse mesmo domínio [13].

O *Domain-Driven Design* é baseado nos seguintes objetivos:

- Colocar o foco principal do projeto no domínio central e na lógica de domínio;
- Basear projetos complexos num modelo do domínio;
- Iniciar uma colaboração criativa entre especialistas técnicos e de domínio para refinar iterativamente um modelo conceitual que aborda problemas de domínio específicos.

3.2. Arquitetura do Portal

O portal, como todos os *Web-Portals* tem uma vertente *frontend* e uma vertente *backend*.

Frontend e *backend* são como referido anteriormente, o módulo responsável pela apresentação da informação e o módulo de armazenamento de dados e de implementação das regras de negócio. Cada vertente precisa de comunicar e funcionar efetivamente com a outra vertente, funcionando assim como uma unidade única para melhorar a funcionalidade do website.

O *frontend* é a parte de um website onde o utilizador interage diretamente. Também é referido como o *cliente-side* da aplicação. Inclui tudo o que os utilizadores experimentam diretamente: cores e estilos de texto, imagens, botões, menus de navegação, etc. O *Developer* deve garantir que o site tenha uma boa capacidade de resposta e que nenhuma parte do site se comporte de forma anormal.

O *backend* é o lado do servidor do website. No *backend* vai-se armazenar e organizar os dados e também garantir que tudo do lado do cliente (*cliente-side*) do website funcione bem. É a parte do website em que o utilizador não consegue ver nem interagir. É também a parte do software que não entra em contato direto com os utilizadores. As partes e características desenvolvidas pelos *developers* de *backend* são acedidas indiretamente pelos utilizadores por meio de uma aplicação *frontend*.

Para a estrutura do Portal Único do Cliente Rangel temos:

Backend:

- *Database*: Onde todos os dados serão guardados, tanto registros de compras, credenciais de utilizadores, etc.;
- *Entity Framework*: Neste caso serve para tornar dados em objetos.
- *Repository*: Irá servir como uma camada abstrata entre a lógica da nossa aplicação e o acesso aos dados;
- *Domain*: Irá servir para a implementação de classes, modelos e o mapeamento da nossa base de dados;
- *WebApi* interna: É uma *API* que pode ser acedida através de protocolos (*HTTP* por exemplo);
- *Swagger*: É uma *API* que ajuda na descrição da língua de programação de forma a descrever *REST API's* (usado para testes).
- *WebApi* externa: Similar a interna, servindo como uma camada extra de proteção, sendo assim mais complicado aceder aos nossos dados, ficando menos comprometidos a certas infrações;

Frontend:

- *Angular*: Ferramenta de *frontend* para criar vistas para clientes através de um browser;

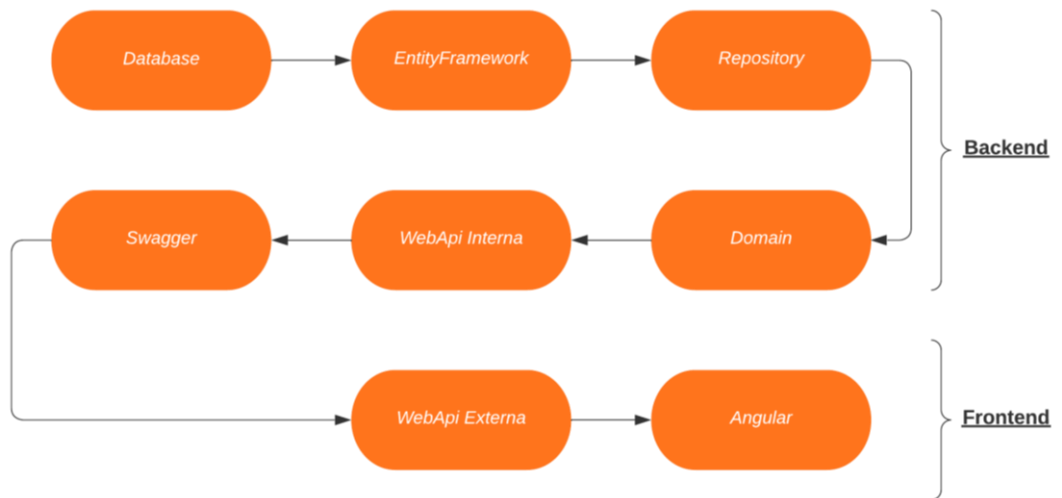


Figura 12 - Camadas da arquitetura do Portal [referência própria]

3.3. Tecnologias usadas no desenvolvimento do Portal

Nesta seção são apresentadas as várias tecnologias usadas para a criação deste portal:

Backend:

- C# (Visual Studio 2019).
- SQL Server Management Studio.
- Tibco.
- Frontend:
 - Angular (VS Code)
 - TypeScript / JavaScript / CSS / HTML (VS Code)
 - TortoiseSVN (Semelhante ao GitHub)

3.3.1. IDE's

Um IDE (*Integrated Development Environment*), é um software para construção de aplicações, que combina ferramentas comuns de desenvolvimento numa única interface gráfica de utilizador. Um IDE normalmente consiste num editor de texto um sistema local de compilação do projeto e de um debugger que ajuda para fazer testes ao código.

Visual Studio 2019

Microsoft Visual Studio é um IDE da Microsoft para o desenvolvimento de software especialmente dedicado ao .NET Framework e às linguagens Visual Basic, C, C++, C#. É um produto de desenvolvimento na área de programação web, usando a plataforma *ASP.NET* para o desenvolvimento de websites, aplicações e serviços web e aplicações móveis.

Neste caso a versão utilizada foi a versão 2019 e foi usada a linguagem C# (C sharp), numa framework *.Net Core Framework*.

Visual Studio Code

O Visual Studio Code é um editor de código-fonte desenvolvido pela Microsoft para Windows, Linux e macOS. Inclui suporte para debugging, controlo de versões de *Git* incorporado, *snippets* e *refactoring* de código, etc. É customizável, permitindo que os utilizadores possam mudar o tema do editor, teclas de atalho e preferências. Um estudo feito no website Stack Overflow o Visual Studio Code foi considerado o IDE mais popular [14].

3.3.2. Linguagens de Programação

Uma linguagem de programação pode ser descrita como qualquer conjunto de regras para instruir um computador ou um dispositivo informático para realizar tarefas específicas.

Uma linguagem de programação é usada principalmente para programar aplicações de computador, *websites* e aplicações móveis.

Existem três tipos de linguagens de programação [15]:

- **Nível Baixo:** Não existe a necessidade de um compilador. Estes programas, baseado em instruções chamadas de código máquina, são diretamente interpretadas pelo *hardware*;
- **Nível Médio:** Este tipo de linguagem de programação fica entre o alto e o baixo. tem como vantagens o suporte de recursos de programação de alto nível, ser uma linguagem amigável e intimamente relacionada com o código máquina e linguagem humana;
- **Nível Alto:** São projetados para programar *websites* e *user-friendly softwares*. Precisam de um compilador;

C#

O C# foi projetado por Anders Hejlsberg da Microsoft em 2000 [16] e mais tarde foi aprovado como padrão internacional pelo Ecma (ECMA-334) em 2002 e ISO (ISO/IEC 23270) em 2003. A Microsoft introduziu o C# junto com o .NET Framework e o Visual Studio.

É uma linguagem multi-paradigma e é uma variante da língua C usada para programar aplicações web, aplicações de computador, aplicações móveis, jogos e muito mais.

SQL

SQL foi inicialmente desenvolvido na IBM por Donald D. Chamberlin e Raymond F. Boyce depois de aprender sobre o modelo relacional de Edgar F. Codd no início dos anos 70 [17].

SQL (Structured Query Language) é uma linguagem específica de domínio usada na programação e projetada para fazer a gestão de dados mantidos numa base de dados. É particularmente útil no tratamento de dados estruturados, ou seja, dados que incorporam relações entre entidades e variáveis.

O SQL oferece duas vantagens principais sobre as *API's* de leitura e gravação. Em primeiro lugar, introduziu o conceito do acesso de muitos registos com um único comando. Em segundo lugar, elimina a necessidade de especificar como alcançar um registo, por exemplo com ou sem índice.

JavaScript

JavaScript muitas vezes abreviado JS, é uma linguagem de programação que é uma das principais tecnologias da *World Wide Web*, juntamente com HTML e CSS. Mais de 97% dos sites usam JavaScript em *client-side* para o comportamento da página da Web [18].

Possui orientação a objetos baseada em protótipos e funções de primeira classe. É multi-paradigma, suportando estilos de programação orientados a eventos. Possui interfaces de programação de *API's* para trabalhar com texto, datas, expressões regulares, estruturas de dados padrão e o *Document Object Model* (DOM).

CSS

CSS (*Cascading Style Sheets*) é uma linguagem que é usada como *styling*, servindo para complementar o HTML [19]. CSS é usado para permitir a separação da apresentação e do conteúdo, incluindo *layout*, cores e fontes, deixando assim o *website* mais apresentável para os seus visitantes. Essa separação pode permitir que várias páginas da web compartilhem a formatação especificando o CSS relevante num arquivo *.css* separado, o que reduz a complexidade e a repetição no conteúdo estrutural. Tem certas vantagens como a sua rapidez e a sua responsividade.

HTML

O *HTML* (*HyperText Markup Language*) não é considerado uma linguagem de programação, mas sim uma linguagem de marcação que serve para documentos serem exibidos num navegador web [20]. Pode ser complementado com o JavaScript e CSS de forma a ter mais funcionalidade e melhorar a funcionalidade do mesmo [21]. O HTML é o componente mais importante de uma aplicação web, visto que é usado na criação da estrutura da mesma. Tem certas vantagens, como a sua integração e a sua portabilidade (funcionando em qualquer navegador).

TypeScript

TypeScript é uma linguagem de programação desenvolvida pela Microsoft em 2012. É um “superconjunto” de JavaScript, adicionando ao mesmo recursos e ferramentas não presentes na sua forma básica [22]. Como o TypeScript é um superconjunto de JavaScript, os programas JavaScript existentes também são programas válidos do TypeScript.

O TypeScript pode ser usado para desenvolver aplicações JavaScript para execução em *client-side* e *server-side* (como no React.js ou Angular.js).

A principal vantagem é ter recursos que são superiores ao JavaScript tal como tipagem estática, corrigir erros em tempo real e descobrir e corrigir erros.

3.3.3. Softwares

SQL Server Management Studio

O *SQL Server Management Studio (SSMS)*, lançado em 2005, é uma aplicação em que é usada para administrar e configurar os componentes do *Microsoft SQL Server*, sendo lançado juntamente com o *Microsoft SQL Server 2005*.

Esta ferramenta inclui editores de *script* e ferramentas que trabalham com objetos e recursos de um servidor [23].

É usada para projetar consultas e fazer a gestão de base de dados e *data warehouses* tanto em *cloud* como remotamente. Algumas das suas vantagens são o seu baixo custo e a sua eficiência.

Tibco ActiveMatrix

TIBCO ActiveMatrix é um *software* lançado pela Tibco Software Inc. que é uma empresa americana de software de inteligência de negócios que foi fundada em 1997 em Califórnia.

TIBCO ActiveMatrix é uma plataforma de tecnologia neutra para aplicações de gestão de processos de negócios e de arquitetura orientada a serviços. A plataforma inclui produtos para a integração e a criação de serviços, serviços distribuídos e de dados. Tem como vantagens um bom controlo e flexibilidade. A principal desvantagem, está relacionada com a pequena comunidade de utilizadores, sendo difícil interagir com especialistas para a resolução de problemas e soluções.

TortoiseSVN

TortoiseSVN é um cliente da Apache subversion (SVN). Foi implementado como uma extensão de *shell* do Microsoft Windows, que ajuda os programadores a fazer a gestão de diferentes versões do código-fonte de seus programas.

Para fazer comparações com alguns programas mais conhecidos, este software tem funções similares ao GitHub, sendo que neste projeto é utilizado para fazer atualizações de soluções. Tem como vantagens ser de uso fácil, gratuito e ter suporte a todos os protocolos de subversões.

TortoiseSVN ganhou o SourceForge.net 2007 Community Choice Award de “Melhor Ferramenta ou Utilitário para Programadores” [24].

3.4. Funcionamento do Portal

3.4.1. Login e menu principal

Ao aceder ao website do portal (<https://my.rangel.com>), surge um ecrã de login. Para entrar no website, é necessário introduzir as credenciais que serão enviadas por e-mail após contacto com a empresa.



Figura 13 - Menu Login [25]

Quando feito o login, é exibido um ecrã de boas-vindas, junto com um painel de controlo. Aqui temos um resumo da informação geral da entidade selecionada, apresenta todos os menus a que a entidade pode aceder e mostra os estados dos envios do dia por localização geográfica.

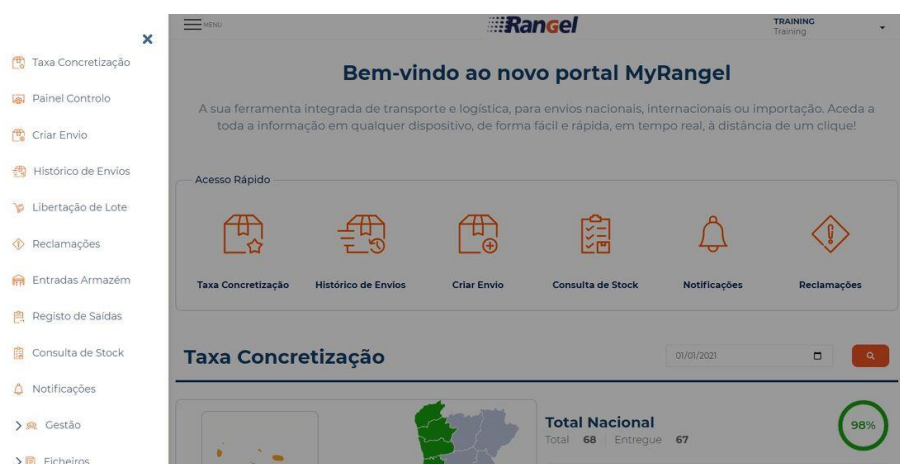


Figura 14 - Menu Principal [25]

3.4.2. Histórico de Envios

O Histórico de Envios permite ver o histórico de envios e de recolhas dessa entidade.

Todos os menus têm um menu de opções avançadas que normalmente permite gerar um ficheiro (.csv ou .xls) do que está a ser apresentado no ecrã. Dependendo da entidade e do menu que estamos a usar poderá ter outras opções.

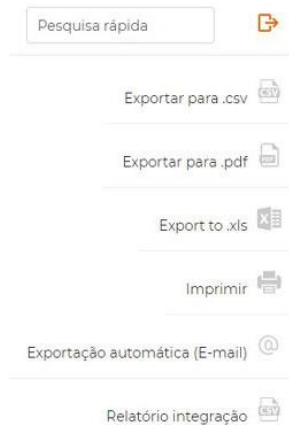


Figura 15 - Opções Avançadas [25]

A pesquisa pode ser feita de várias formas. Na forma de pesquisa normal pode ser feito através da data de envio, sendo apresentada na forma de um calendário, ou através da referência do envio.

Na forma de um calendário, é possível selecionar um intervalo predefinido ou dia do mês de início e fim que incluam os registos, que tenham sido criados no intervalo selecionado.

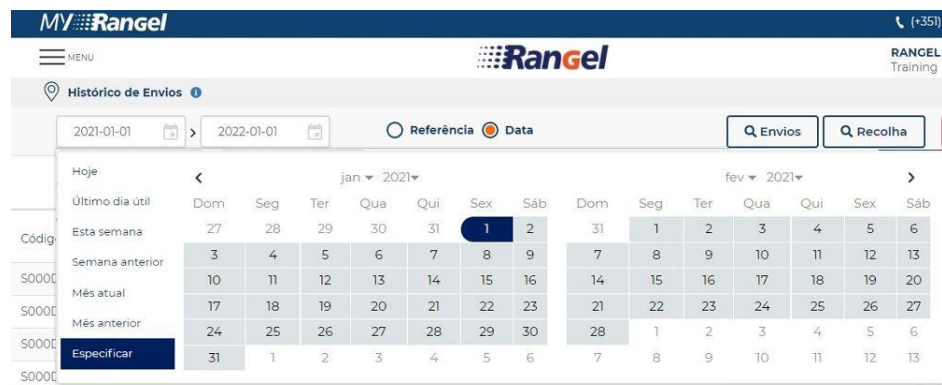


Figura 16 - Pesquisa em calendário [25]

Outro tipo de pesquisa é a pesquisa avançada, que permite ao utilizador definir critérios de pesquisa avançados. Esta pesquisa avançada também está disponível para os restantes menus, mas mudando de opções.

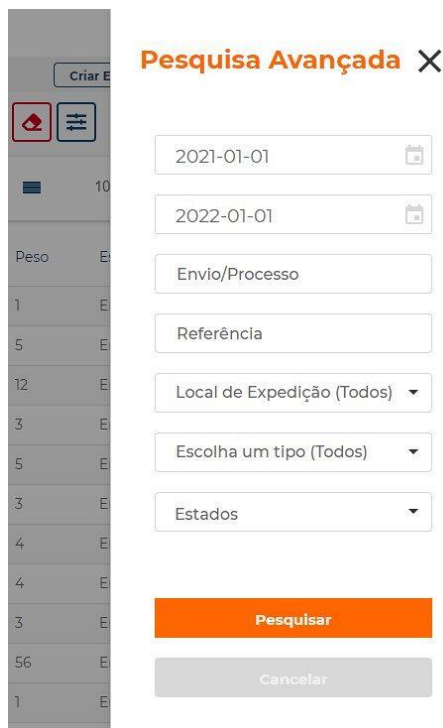


Figura 17 - Pesquisa Avançada [25]

Existem também ações, que normalmente estão ocultas e apenas um utilizador com determinado tipo de privilégios pode aceder, que permitem modificar os dados de um determinado envio

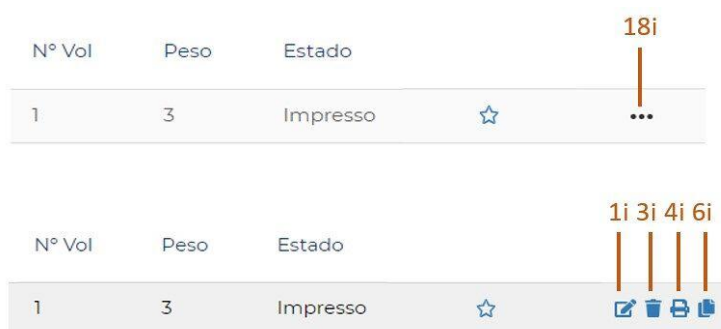


Figura 18 - Ações avançadas (1) [25]

A exibição das ações que permitam consultar os dados de um determinado envio dependem do estado do mesmo. O exemplo que aqui se descreve, exhibe todas as ações de consulta de um envio, pois o mesmo encontra-se no estado 'Entregue' há mais de 48 horas.

Nº Vol	Peso	Estado	9i	8i	7i	10i	2i
1	3	Entregue					

Figura 19 - Ações avançadas (2) [25]

Na forma de um botão, dependendo do perfil do utilizador, é possível determinar a localização da viatura no mapa de distribuição nacional e o registo de temperatura ambiente do veículo. Ambas as funções estão restritas a envios que estejam no estado diferente de ‘Criado’, ‘Impresso’ ou ‘Encerrado’.

A aba de ‘Informação da Viagem’ permite ver os pontos de paragem na rota do veículo, enquanto a aba ‘Temperatura Ambiente de Veículo’ exhibe o registo de leitura de todos os sensores equipados no veículo.

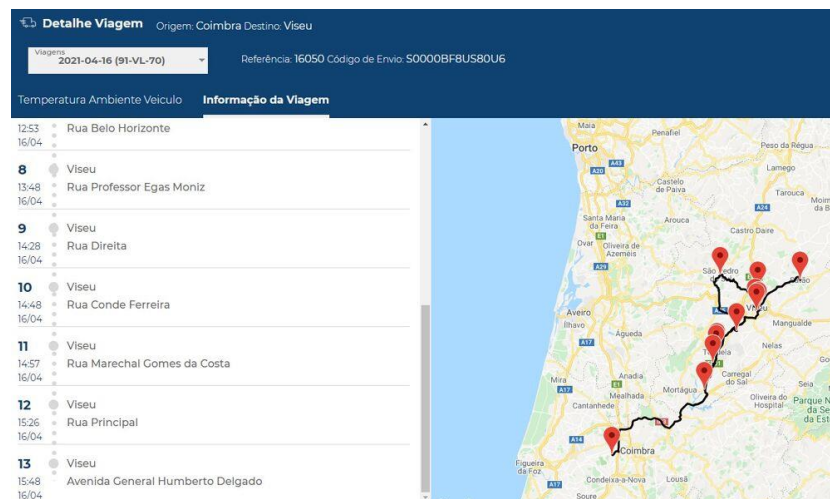


Figura 20 - Informação da Viagem [25]



Figura 21 - Temperatura Ambiente de Veículo [25]

Outras opções, permitem ao utilizador configurar os meios de notificação automática para o envio selecionado, seja por SMS ou o envio do POD (*Proof of Delivery*/Prova de Entrega). Também é possível o utilizador consultar e descarregar o POD assim que disponível. Pode ainda consultar e descarregar a GR (Guia de Remessa), editar e efetuar alterações a um registo criado, eliminar definitivamente o registo de envio selecionado, duplicar e transformar a informação do registo selecionado e consultar a informação do envio.

3.4.3. Criar envio

Neste menu é possível criar um envio. O envio está dividido em 4 partes. A primeira decorre no expedidor onde os dados do local de expedição serão automaticamente preenchidos com os dados da entidade configurada, sendo permitido editar esses dados. O campo 'Remetente' exibe todos os endereços que tenham sido gravados como local de expedição para tornar o preenchimento dos dados do local de recolha automatizado.

01. Expedidor Editar

Guardar contacto no livro de endereços

Expedidor	Contacto
Remetente LocalExp	Nome Contacto Teste
* País Portugal	* Email training@rangel.com
Empresa LocalExp	Telefone 220000001
Endereço RUA DA SERRA - FOLGOSA APARTADO 119	Telemóvel 960000001
Código Postal 4446-909	* Data da Recolha 01/01/2021
* Cidade Maia	* Hora carga pronta 09:00
* NIF 50000001	* Hora fecho 18:00

Figura 22 - Criação de Envio (Expedidor) [25]

Na segunda parte são preenchidos os dados de envio, tais como os dados do destinatário ou local de entrega do envio e o tipo de serviço de transporte.

Figura 23 - Criação de Envio (Dados Envio) [25]

A terceira parte caracteriza os volumes a transportar, sendo possível selecionar os tipos de volume e o tipo de embalagem.

Figura 24 - Criação de Envio (Volumes) [25]

A quarta parte trata a parte documental, sendo possível efetuar o envio de documentos tais como a fatura ou outros documentos complementares à expedição.

Figura 25 - Criação de Envio (Volumes) [25]

Após estes quatro momentos, será feita a validação de todos os campos preenchidos e obrigatórios. O utilizador verá um pop-up com o progresso de validação até que o envio seja validado e criado.

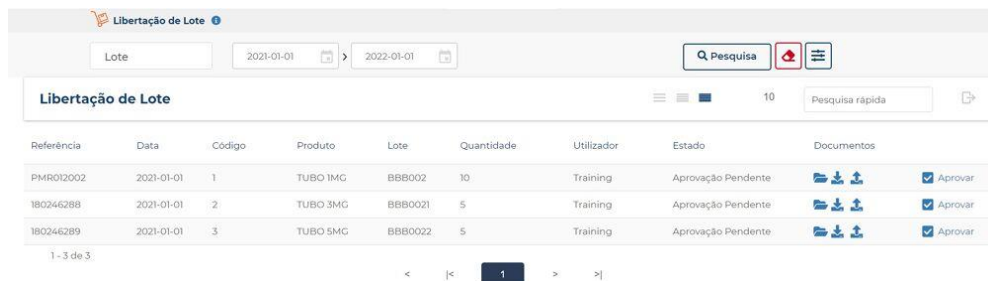
Assim que o registo é criado, o utilizador será reencaminhado para o ecrã 'Histórico de envios', onde o registo será guardado como temporário e com o estado 'Criado'.

Assim que um novo envio tenha sido registado e esteja disponível no ecrã 'Histórico de envios' será possível criar uma etiqueta e fazer a impressão do manifesto.

3.4.4. Libertação de Lote

Na libertação de lote é possível fazer a consulta de pedidos de Libertação de Lote, assim como a sua aprovação. Por forma a garantir a segurança no processo de aprovação, foram implementados dois mecanismos de segurança, divididos por níveis de acesso.

O primeiro nível de acesso, garante quais os utilizadores que estão autorizados a consultar as fichas de libertação de lote. O segundo nível de acesso é implementado através da verificação de coordenadas do cartão de utilizador, no momento da aprovação de libertação de lote.



Referência	Data	Código	Produto	Lote	Quantidade	Utilizador	Estado	Documentos
PMR012002	2021-01-01	1	TUBO 1MG	BBB002	10	Training	Aprovação Pendente	<input checked="" type="checkbox"/> Aprovar
180246288	2021-01-01	2	TUBO 3MG	BBB0021	5	Training	Aprovação Pendente	<input checked="" type="checkbox"/> Aprovar
180246289	2021-01-01	3	TUBO 5MG	BBB0022	5	Training	Aprovação Pendente	<input checked="" type="checkbox"/> Aprovar

Figura 26 - Libertação de Lote [25]

É permitido ao utilizador consultar os documentos anexo ao pedido de libertação de lote, descarregar a ficha de libertação de lote e anexar documentos no pedido ficha de libertação de lote.

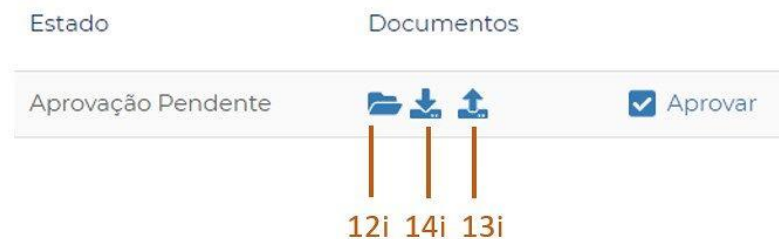
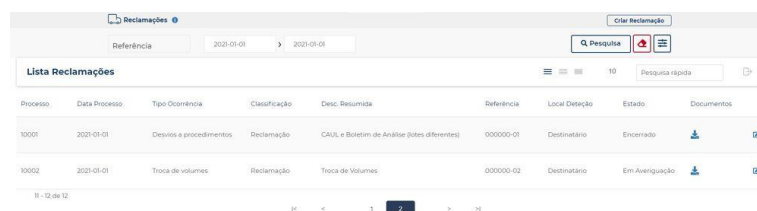


Figura 27 - Opções Avançadas (3) [25]

3.4.5. Reclamações

Neste menu o utilizador pode consultar, editar e efetuar alterações a um registo de reclamação criado.



Processo	Data Processo	Tipo Ocorrência	Classificação	Desc. Resumida	Referência	Local Deteção	Estado	Documentos
10001	2021-01-01	Deposito e procedimentos	Reclamação	CAUI, e Boletim de Análise (ótes diferentes)	000000-01	Destinatário	Encerrado	<input checked="" type="checkbox"/>
10002	2021-01-01	Troca de volumes	Reclamação	Troca de Volumes	000000-02	Destinatário	Em Aversigação	<input checked="" type="checkbox"/>

Figura 28 – Reclamações [25]

3.4.6. Entrada Armazém

Este menu permite ao utilizador a consulta e gestão do produto até ao momento de chegada ao armazém. Para além de exibir o estado, também é possível descarregar a informação do CMR (documento de acompanhamento).



Entradas Armazém						
Nº Documento	Data Doc.	Ref. Fornecedor	Fornecedor	Operação	Estado	Download
01/01/2021			Rangel LX	Entrada-Fornecedores	Arrumado	
Ref. Fornecedor	Cód. Produto	Produto	Quantidade	Quantidade Doc.	Preço	Estado
	5016407	PROD TEST 1	1000	1000		Arrumado
	5016423	PROD TEST 2	1000	1000		Arrumado

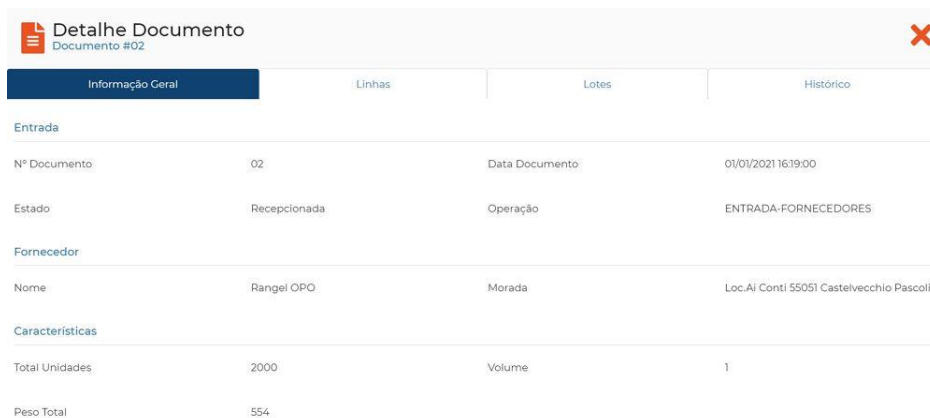
Figura 29 - Entradas Armazém [25]

À semelhança do menu ‘Histórico de Envios’, este menu também tem opções avançadas para fazer o download de ficheiros .cls e .xls, também permite fazer as mesmas formas de pesquisa normal e avançada, acrescentando ainda uma opção de pesquisa rápida que quando introduzido texto neste campo, exibe a informação do produto em que tenha sido encontrado um valor que coincida em qualquer uma das colunas do ecrã. Por exemplo, no resultado de pesquisa com mais de 100 registos, se introduzido um produto, será exibido o registo específico.



Figura 30 - Pesquisa Rápida [25]

Ainda é possível ao utilizador aceder ao detalhe dos documentos.

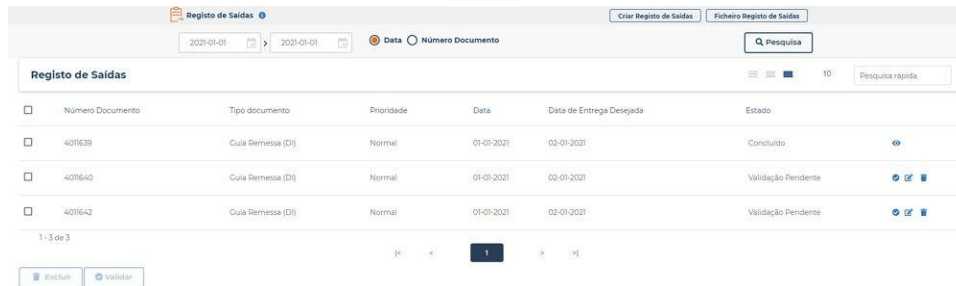


Detalhe Documento			
Documento #02			
Informação Geral	Linhas	Lotes	Histórico
Entrada			
Nº Documento	02	Data Documento	01/01/2021 16:19:00
Estado	Recepcionada	Operação	ENTRADA-FORNECEDORES
Fornecedor			
Nome	Rangel OPO	Morada	Loc.Ai Conti 55051 Castelvecchio Pascoli
Características			
Total Unidades	2000	Volume	1
Peso Total	554		

Figura 31 - Detalhe Documento [25]

3.4.7. Registo de Saídas

Este menu permite ao utilizador consultar, validar e criar pedidos de saídas. De forma a assegurar o registo correto de um pedido de saída, mesmo depois de criado, a submissão do mesmo no sistema obriga à validação manual sobre a forma de introdução de coordenadas do cartão do utilizador.

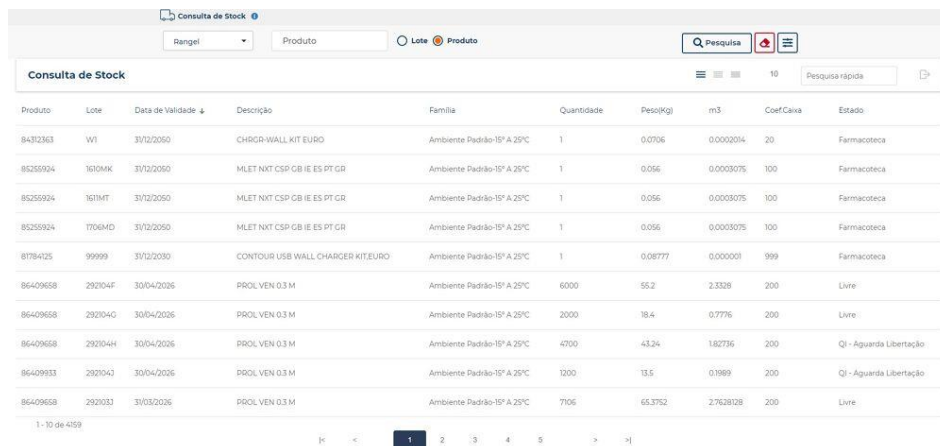


Número Documento	Tipo documento	Prioridade	Data	Data de Entrega Desejada	Estado
401639	Guia Remessa (DI)	Normal	01-01-2021	02-01-2021	Concluído
401640	Guia Remessa (DI)	Normal	01-01-2021	02-01-2021	Validação Pendente
401642	Guia Remessa (DI)	Normal	01-01-2021	02-01-2021	Validação Pendente

Figura 32 - Registo de Saídas [25]

3.4.8. Consulta Stock

Este ecrã permite ao utilizador a consulta e gestão de disponibilidade de produto no armazém. Tem as mesmas opções avançadas que os outros menus já apresentados.



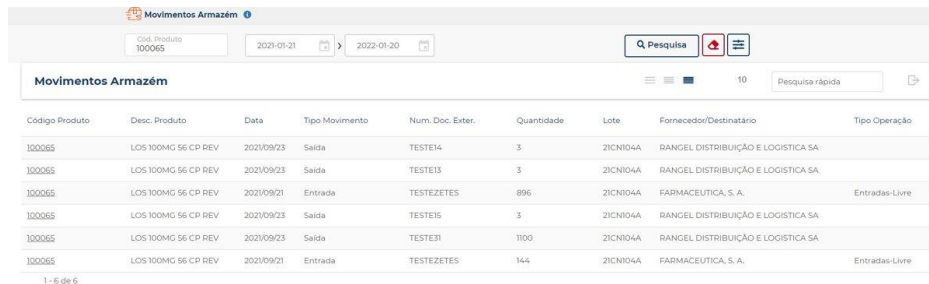
Produto	Lote	Data de Validade	Descrição	Família	Quantidade	Peso(Kg)	m3	Coef.Caixa	Estado
8432363	WI	31/12/2050	CHDGR-WALL KIT EURO	Ambiente Padrão-15° A 25°C	1	0,0706	0,0002014	20	Farmacoteka
8525924	1610MK	31/12/2050	MLET NXT CSP GB IE ES PT GR	Ambiente Padrão-15° A 25°C	1	0,056	0,0003075	100	Farmacoteka
8525924	1611MT	31/12/2050	MLET NXT CSP GB IE ES PT GR	Ambiente Padrão-15° A 25°C	1	0,056	0,0003075	100	Farmacoteka
8525924	1706MD	31/12/2050	MLET NXT CSP GB IE ES PT GR	Ambiente Padrão-15° A 25°C	1	0,056	0,0003075	100	Farmacoteka
878425	99999	31/12/2030	CONTOUR USB WALL CHARGER KIT,EURO	Ambiente Padrão-15° A 25°C	1	0,08777	0,000001	999	Farmacoteka
86409658	29204F	30/04/2026	PROL VEN 0,3 M	Ambiente Padrão-15° A 25°C	6000	55,2	2,3328	200	Livre
86409658	29204G	30/04/2026	PROL VEN 0,3 M	Ambiente Padrão-15° A 25°C	2000	18,4	0,7776	200	Livre
86409658	29204H	30/04/2026	PROL VEN 0,3 M	Ambiente Padrão-15° A 25°C	4700	43,24	1,82736	200	QI - Aguarda Libertação
86409658	29204J	30/04/2026	PROL VEN 0,3 M	Ambiente Padrão-15° A 25°C	1200	13,5	0,1989	200	QI - Aguarda Libertação
86409658	29203J	31/03/2026	PROL VEN 0,3 M	Ambiente Padrão-15° A 25°C	7106	65,3752	2,7628028	200	Livre

Figura 33 - Consulta Stock [25]

Em termos de pesquisa, é possível procurar e filtrar na lista de stock por Lote ou Produto. Também tem a pesquisa rápida e a pesquisa avançada igual ao menu 'Entradas Armazém'.

3.4.9. Movimentos Armazém

Este menu permite ao utilizador a consulta e gestão de movimentos de produto no armazém. A informação está disposta por código de produto, com a descrição do produto e detalhe de quantidade e coeficiente de caixa. Tem as mesmas opções avançadas do que os outros painéis.



The screenshot shows the 'Movimentos Armazém' interface. At the top, there is a search bar with 'Cód. Produto 100065' and date filters for '2021-01-21' and '2022-01-20'. Below the search bar, the title 'Movimentos Armazém' is displayed. The main content is a table with the following columns: Código Produto, Desc. Produto, Data, Tipo Movimento, Num. Doc. Exter., Quantidade, Lote, Fornecedor/Destinatário, and Tipo Operação. The table contains six rows of data for product 100065, showing various movements like 'Saída' and 'Entrada' with specific dates and quantities. At the bottom left, it indicates '1 - 6 de 6'.

Código Produto	Desc. Produto	Data	Tipo Movimento	Num. Doc. Exter.	Quantidade	Lote	Fornecedor/Destinatário	Tipo Operação
100065	LOS 100MG 56 CP REV	2021/09/23	Saída	TESTE14	3	21CN104A	RANGEL DISTRIBUIÇÃO E LOGISTICA SA	
100065	LOS 100MG 56 CP REV	2021/09/23	Saída	TESTE13	3	21CN104A	RANGEL DISTRIBUIÇÃO E LOGISTICA SA	
100065	LOS 100MG 56 CP REV	2021/09/21	Entrada	TESTEZETES	896	21CN104A	FARMACEUTICA, S. A.	Entradas-Livre
100065	LOS 100MG 56 CP REV	2021/09/23	Saída	TESTE15	3	21CN104A	RANGEL DISTRIBUIÇÃO E LOGISTICA SA	
100065	LOS 100MG 56 CP REV	2021/09/23	Saída	TESTE31	1100	21CN104A	RANGEL DISTRIBUIÇÃO E LOGISTICA SA	
100065	LOS 100MG 56 CP REV	2021/09/21	Entrada	TESTEZETES	144	21CN104A	FARMACEUTICA, S. A.	Entradas-Livre

Figura 34 - Movimentos Detalhe [25]

Na pesquisa é obrigatório procurar e filtrar na lista de movimentos por código de produto e no limite de intervalo de datas até 365 dias. Também tem pesquisa rápida e pesquisa avançada.

Ainda é possível ao utilizador aceder ao detalhe do produto e desta forma consultar informação dispersa pelas várias abas selecionadas no 'Código do produto' da lista.

3.4.10. Produtos

Este menu permite ao utilizador a consulta detalhada do produto no armazém. A informação está disposta por código de produto com a descrição do produto e detalhe sobre peso, unidade por caixa e dimensões.



The screenshot shows the 'Produtos' interface. At the top, there is a search bar with 'Cód. Produto 100065' and date filters for '2021-01-21' and '2022-01-20'. Below the search bar, the title 'Produtos' is displayed. The main content is a table with the following columns: Código Produto, Desc. Produto, Família, Peso, Un. por caixa, No. de unidades por palete, and Dimensões. The table contains one row of data for product 100065, showing details like 'Ambiente - Standard-15°C a 25°C', weight of 60, 56 units per box, 896 units per pallet, and dimensions of 9 x 8 x 4. At the bottom left, it indicates '1 - 1 de 1'.

Código Produto	Desc. Produto	Família	Peso	Un. por caixa	No. de unidades por palete	Dimensões
100065	LOS 100MG 56 CP REV	Ambiente - Standard-15°C a 25°C	60	56	896	9 x 8 x 4

Figura 35 – Produtos [25]

Tudo acerca das diferentes pesquisas é igual ao menu 'Movimentos de Armazém'.

3.4.11. Notificações

Entre as várias opções, é possível configurar o envio da notificação de eventos de armazenagem e distribuição por SMS ou email e até o envio do POD assim que disponível.

Criar Notificação

Descrição: Formação

Telemóvel

Email To: formacao@rangel.com

Email CC

Email BCC

picking Completo: SMS Email

Saída Armazém: SMS Email

Entrega: SMS Email

Incidências: SMS Email

POD:

Cancelar

Figura 36 - Criação de Notificação (1) [25]

Criar Notificações

Notificações

Adicionar

			Picking Completo	Saída do Armazém	Entrega	Incidências						
<input type="checkbox"/>	Embalagem	Email	Telemóvel	SMS	EMAIL	SMS	EMAIL	SMS	EMAIL	SMS	EMAIL	POD
<input type="checkbox"/>	Formação	formacao@rangel.com	●	●	●	●	●	●	●	●	●	●

Figura 37 - Criação de Notificação (2) [25]

3.4.12. Gestão

A categoria Gestão concentra as subcategorias Livro de Endereços e Embalagens e será onde o utilizador poderá consultar, criar, editar e importar os endereços que irão compor o Livro de Endereços e também consultar, criar, editar as embalagens com características personalizadas no ecrã Embalagens.



Figura 38 - Painel Gestão [25]

3.4.13. Livro de Endereços

Neste menu é possível fazer a consulta e gestão dos endereços gravados como local de recolha ou destino. Os endereços que tenham sido criados a partir do ecrã ‘Criar envio’ quando seleccionada a opção ‘Guardar contacto no livro de endereços’ também estarão disponíveis.

Código	Empresa	Endereço	NIF	Nome Contacto	Telefone	Telemóvel	E-mail	Localidade
0001	TRAINING RGL1 Lda	Rua Rangel		TRAINING RGL1	220000001	960000001		Maia
100000	LocalEip	RUA DA SERRA - FOLGOSA APARTADO 119	50090008	Training	220000001	960000001	training@rangel.com	Maia

Figura 39 - Livro de Endereços [25]

3.4.14. Embalagens

Este menu permite a consulta e gestão das embalagens com características personalizadas. O nome das embalagens que tenham sido criados a partir do ecrã atual estarão disponíveis no ecrã ‘Criar envio’ quando seleccionada a opção ‘Embalagem’ no sector ‘Volumes’.

Embalagem	Comprimento	Largura	Altura
BOX Type 2	30	20	5
BOX Type 1	45	20	5
BOX Type 3	20	20	5

Figura 40 – Embalagens [25]

3.4.15. Criar Embalagem

Este menu permite ao utilizador adicionar uma embalagem e gravar com um nome único.

3.5. Contribuições no âmbito do estágio

Neste capítulo apresentamos as contribuições do autor para as melhorias do Portal da Rangel e outros projetos em que o autor participou. Apesar de o autor ter participado em bastantes contribuições, uma grande parte foram projetos pequenos ou até mesmo correção de erros e melhorias gerais. Assim apresentamos alguns projetos maiores em que o autor participou e o progresso desde o primeiro projeto até ao último, mencionando apenas os mais importantes em termos de relevância para o projeto e para a aprendizagem do autor.

3.5.1. Primeiro Projeto

Procura do envio na base de dados do *GCOM*

GCOM é um sistema de gestão de armazéns.

O problema consistia no seguinte: ao procurar por um envio este podia estar na base de dados interna da Rangel ou na base de dados do *GCOM* e o *MyRangel* apenas tinha sido configurado para retornar dados da base de dados interna da Rangel.

A primeira intervenção consistiu na reformulação do processo da criação de uma condição em que se não tivesse esse envio na base de dados interna da Rangel, ir-se-ia procurar na base de dados do *GCOM* e de seguida mudava-se a função usada para receber os dados corretos. Finalmente, o autor interagiu com o *frontend*, mapeando os dados recebidos na página a ser mostrada aos clientes da Rangel.

O processo pode resumir-se da seguinte forma:

Primeiro começa-se por introduzir o número de encomenda do portal, de seguida:

1. Através do Angular ele chama o serviço *HTTP* onde vai fazer uma chamada *API*.
2. Continua esta chamada *API* para o *backend*.
3. O *backend* faz uma chamada à base de dados (neste caso a uma *stored procedure*).

Em função da resposta:

- **SIM:** Caso os dados existam na base de dados interna da Rangel esta vai retornar os dados necessários
- **NÃO:** Caso os dados não existam na base de dados interna da Rangel, a pesquisa vai-se fazer na base de dados do *GCOM*.

De seguida:

1. No *backend*, *parse* (processamento) e mapeamento dos dados retornados pela base de dados.
2. Envio dos dados o *frontend* onde vão ser mapeados para os respetivos campos.

3. Mostrar os dados na *webpage*.

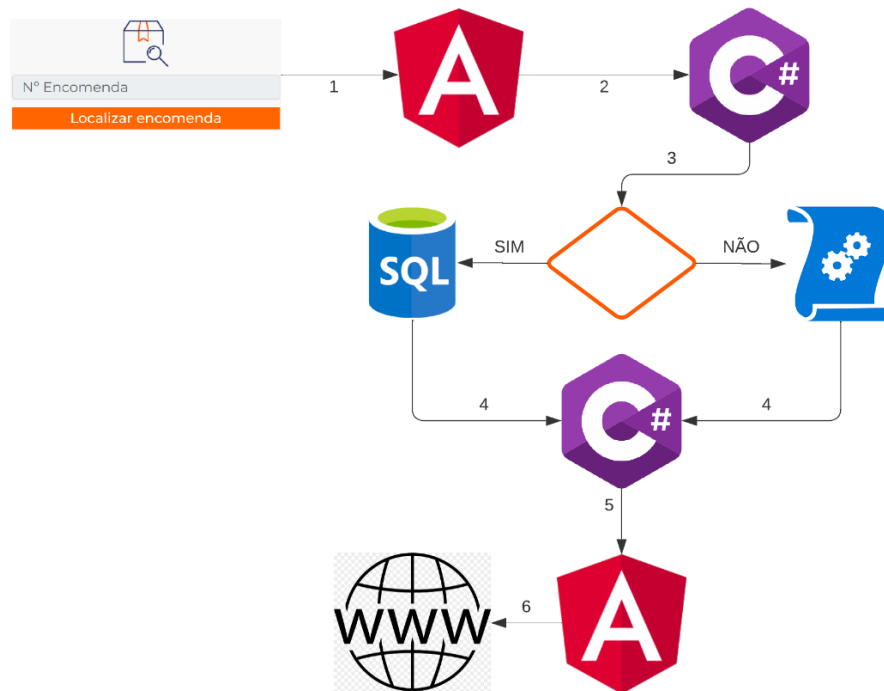


Figura 41 - Processo GCOM [referência própria]

Este primeiro projeto revelou-se uma excelente escolha do coordenador de equipa, visto que nele o autor trabalhou um pouco em todas as partes do sistema, *SQL*, *backend (.net core)* e *frontend (angular)*. Deste modo, o autor começou a analisar e a aprender um pouco destas tecnologias que foram usadas durante toda a duração do estágio.

3.5.2. Projetos intermédios e resolução de bugs

Gerar credenciais ao criar um novo usuário

Como este portal é um portal corporativo, os utilizadores não têm a liberdade de criarem uma conta. Os clientes têm de pedir a criação de uma conta e os administradores do portal irão criar e enviar as credenciais ao cliente. Este processo era feito manualmente e de forma demorada. Nesta intervenção, o autor automatizou o processo. Desenvolveu a criação automática de e-mail com as credenciais do utilizador para serem enviadas aos clientes. Com esta automatização o utilizador ao clicar apenas em alguns botões cria conta do utilizador, uma password aleatória é gerada e um e-mail é enviado para o cliente com as credenciais.

Após algumas discussões a proposta aceite foi a presente na figura 52.

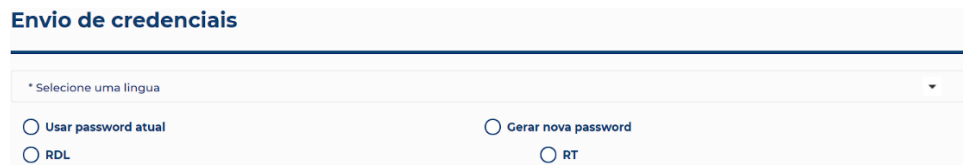


Figura 42 - Envio de credenciais [25]

Algumas especificidades mais foram acauteladas:

1. Visto que o portal está disponível em Português e Inglês criou-se a opção de escolha da língua.
2. Caso fosse pedida alguma password pelo cliente, esta pode ser introduzida manualmente escolhendo-se a opção “Usar password atual”; caso este pedido não fosse pedido será gerado uma password aleatória através da opção “Gerar nova password”.
3. *RDL* (Rede distribuição logística) | *RT* (Rede Transitária). Esta opção foi necessária porque os clientes podem estar associados a entidades diferentes o que faz com que o e-mail tenha campos diferentes, como por exemplo os contactos.

Por fim ao criar a conta ao cliente o processo vai fazer uma chamada *HTTP* em que vai ser evocado um modelo de e-mail previamente criado e os dados são pré preenchidos em função das opções escolhidas anteriormente.

Criar roles e grupo de roles

No painel de criação de um utilizador tem uma opção de dar permissões aos clientes (figura 53), estas permissões servem como restrições para poder aceder apenas a certos menus e executar certas funções permitidas.

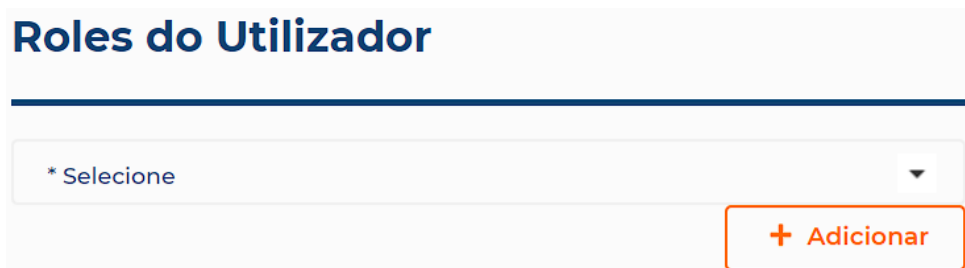


Figura 43 - Roles de utilizador (antigo) [25]

Um exemplo é a *role* “*CreateShipping*” que dá acesso a criação de um envio, e assim tornar possível aceder a esse painel. O problema é que existem muitas *roles* e com o tempo vão existir ainda mais. Para automatizar este processo criaram-se grupos de *roles* para associar a entidades de características semelhantes. Assim, será mais rápido adicionar um utilizador

a um grupo de *roles*, em vez de estar a adicionar cada funcionalidade individualmente. Assim, na opção de escolher cada role individualmente é agora possível escolher uma entidade e associar grupo de *roles*.

Este processo foi feito através de chamadas *SQL*, onde se procura estes grupos e os seus respetivos roles em tabelas previamente criadas, utilizando dados de modelo *one-to-many* (um role para vários grupos).

Foi criada uma *dropdown* nova com os grupos das permissões, e dois novos botões.

O botão “Adicionar grupo”, incrementa os roles aos já existentes, enquanto o botão “Substituir roles” elimina todos os roles já associados ao utilizador e acrescenta apenas os novos roles.

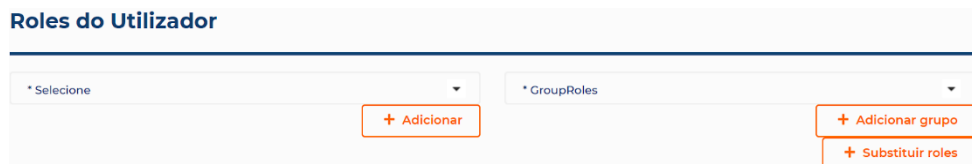


Figura 44 - Roles de utilizador (novo) [25]

Entre os projetos onde o autor interveio surgiam sempre pequenos ajustes que eram necessários para a melhoria do portal, normalmente estes ajustes foram correção de erros ou alguma melhoria pedida pelos clientes da Rangel, através de um *helpdesk* (serviço de apoio aos clientes para suporte e resolução de problemas) disponível.

3.5.3. Projeto Alcotrans

O Projeto da Alcotrans foi o projeto maior em que o autor participou, sendo também o mais importante em termos do nível de utilidade pela empresa. Também foi o projeto com maior nível de dificuldade, porque ao contrário dos outros projetos, foi utilizada uma ferramenta nova, o *Tibco*, do qual o autor não tinha qualquer experiência.

A Alcotrans é uma agente transitória que começou a trabalhar com a Rangel. Foi necessário fazer uma integração dos serviços da Alcotrans, com o portal da Rangel. Identificamos 3 tarefas importantes a serem desenvolvidas para esta integração e todas estas tarefas foram desenvolvidas, testadas e executadas através da ajuda das ferramentas do *Tibco* e do *SQL*.

Primeira tarefa

A primeira tarefa foi a integração de ficheiros. Foi necessário ter um ficheiro com os detalhes todos de um envio, sendo que, foi necessário fazer um

parsing (processamento) dos dados, inseri-los na base de dados interna da Rangel e criar um ficheiro com marcação *.xml*. Este ficheiro tem certos dados específicos pedidos pela Alcotrans. O nome deste ficheiro é o Manifesto.

Processo:

Primeiro é recebido um ficheiro com os dados e detalhes do envio, e de seguida é feito um mapeamento de forma a fazer um processamento desses dados.

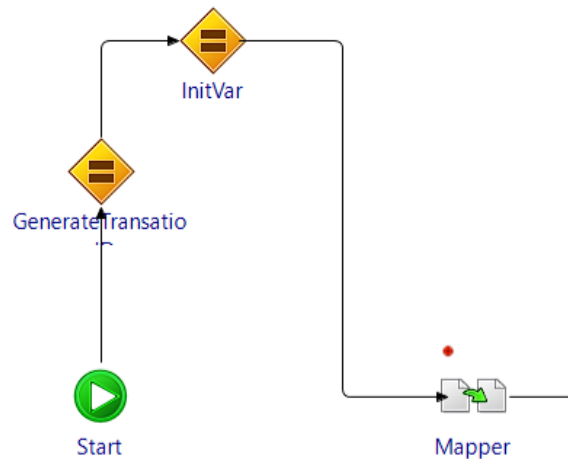


Figura 45 - Mapping dos dados [referência própria]

De seguida é feito o *render* (composição de imagem) dos ficheiros com o *output* do mapeamento da figura 45. Este *render* transforma os dados do elemento do mapeamento (*string de bytes*) numa *string XML*.

Depois, através de duas *stored procedures* diferentes é criado o nome do ficheiro a ser gerado (*GetFileName*, figura 46) e é feita a recolha de dados sobre o serviço a ser utilizado (neste caso serão credenciais da Alcotrans a serem usadas mais tarde).

Por último é transformada a *string XML* num ficheiro com um estilo *XML* (*TransformXML*, figura 46).

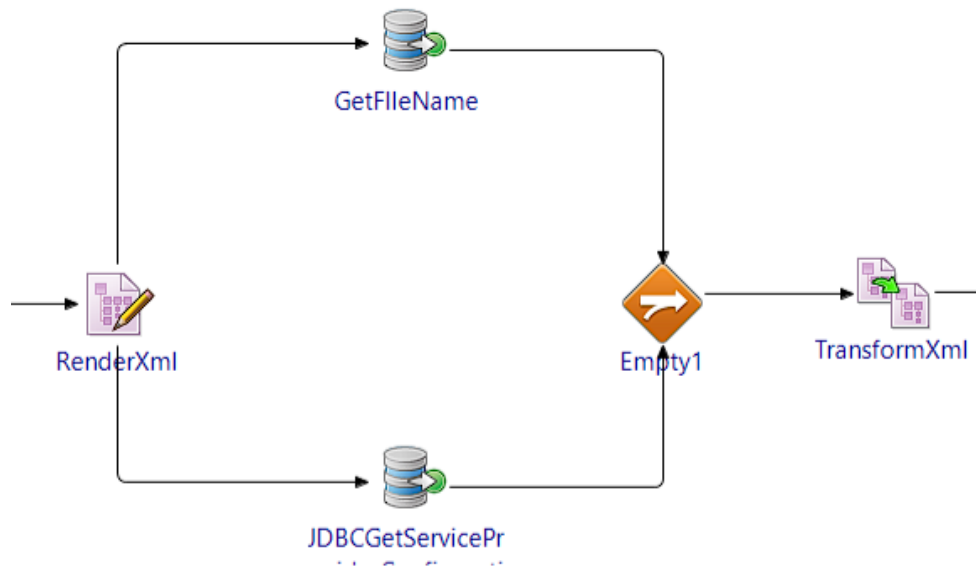


Figura 46 - Render e gerar XML [referência própria]

A última parte da primeira tarefa foi a escrita dos dados da *string .xml* para um ficheiro (*WriteFile*, figura 47) e fazer o *upload* para um sistema da Rangel através do *FTPPUT* (figura 47) com as credenciais retiradas no processo *JDBCGetServicePr* (figura 46). O sistema da Rangel para transferência dos ficheiros tem o nome de *FTP* e foi usado ao longo de todo este processo.

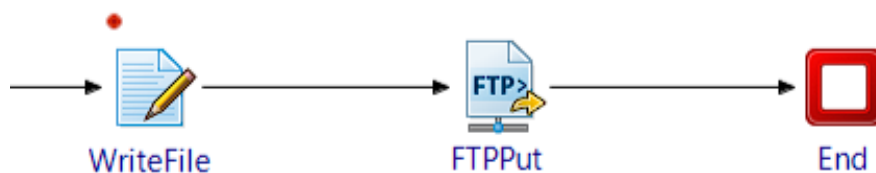


Figura 47 - Escrita e upload do ficheiro [referência própria]

Segunda tarefa

A segunda tarefa foi o *Track and Trace*. Neste processo foram recebidos ficheiros da Alcotrans sobre os vários envios que foram integrados na **primeira tarefa**. Estes ficheiros têm dados sobre os envios, mas dados diferentes dos recebidos da primeira tarefa, visto que estes dados são sobre o processamento do envio, as suas localizações, alterações, etc... O objetivo foi fazer o processamento dos dados dos ficheiros recebidos e guardá-los na base de dados para depois poderem ser utilizados e visualizados pelos nossos clientes através do *MyRangel*.

O **primeiro passo** foi obter as credenciais de login da Alcotrans (*JDBCGetServicePr*, figura 48) e de seguida extrair os ficheiros enviados pela Alcotrans (*FTPDir*, figura 48).

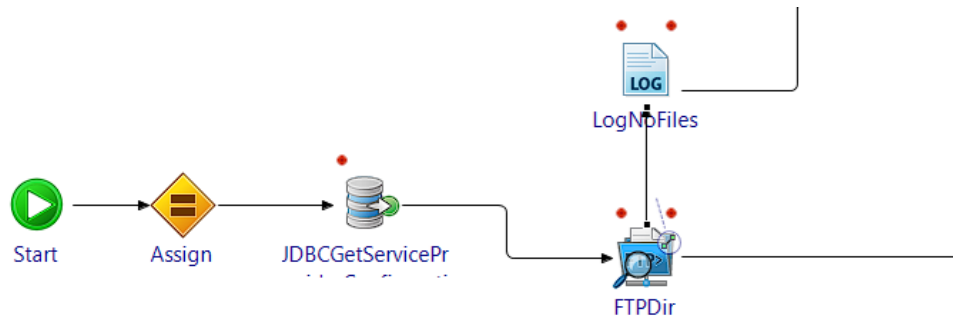


Figura 48 - Obter ficheiros de Track and Trace [referência própria]

De seguida entra-se num ciclo onde para cada ficheiro:

1. Obter o ficheiro e guardá-lo localmente (*FTPGet*, figura 49);
2. Ler o ficheiro de forma a obter os dados (*ReadFile*, figura 49);
3. Fazer o processamento do *output* do *ReadFile* (*ParseXml*, figura 49).

Cada ficheiro pode conter vários envios e cada envio pode conter vários eventos, que mostram as alterações que ocorrem associados a esses envios (normalmente são mudanças de temperatura).

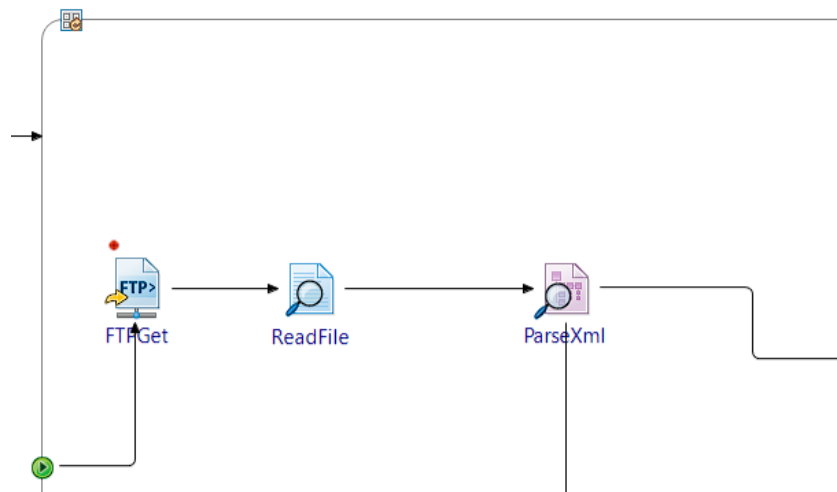


Figura 49 - Ciclo de parse do ficheiro [referência própria]

Ainda dentro do primeiro ciclo entra-se dentro de um segundo ciclo para cada ficheiro:

1. Criar/atualizar esse envio na base de dados (*JDBCCreateShippi*, figura 50);
2. Entrar num terceiro ciclo onde vamos guardar na base de dados as atualizações dos envios (*JDBCCreateTemp*, figura 50).
3. Sair do terceiro ciclo e do segundo ciclo e eliminar o ficheiro lido do *FTP* (*FTPDeleteFile*, figura 50).

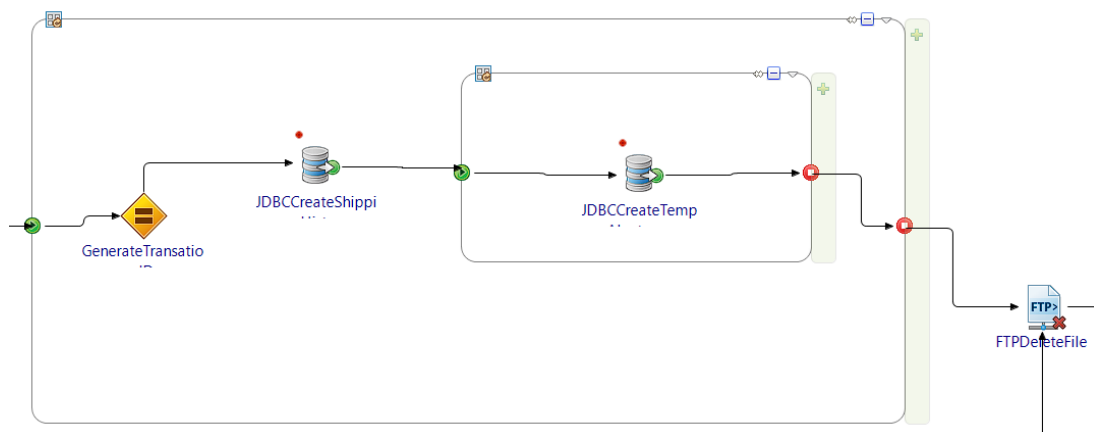


Figura 50 - Criação/Update envios e eventos [referência própria]

Terceira tarefa

A terceira e última tarefa é referente ao *Proof-Of-Delivery* (POD). O *Proof-Of-Delivery* é um ficheiro que é a prova de entrega do envio, finalizando o processo. O objetivo desta última tarefa é receber este documento da parte da Alcotrans e integrá-lo no sistema da Rangel para depois poder ser usado de forma a estar disponível no *frontend* para os clientes da Rangel poderem fazer o download do *Proof-Of-Delivery*.

No início do processo foi preciso obter as credenciais de login da Alcotrans através do processo (*JDBCGetServicePr*, figura 51) e de seguida ir buscar os ficheiros enviados pela Alcotrans (*FTPDir*, figura 51).



Figura 51 - Obter ficheiros de Proof-Of-Delivery [referência própria]

De seguida entramos num ciclo onde para cada *Proof-Of-Delivery* vamos:

1. Obter o ficheiro e guardá-lo localmente (*FTPGet*, figura 52);
2. Obter as informações do ficheiro guardadas na base de dados interna da Rangel através de uma *stored procedure* (*GetPODInfo*, figura 52);
3. Ler o ficheiro de forma a obter dados (*ReadFile*, figura 52);

4. Fazer o mapeamento do *output* do *ReadFile* (*Mapper*, figura 52).

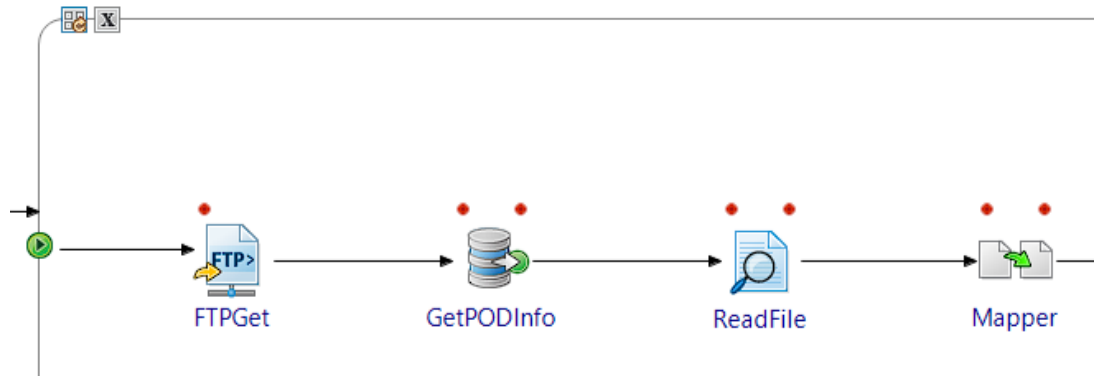


Figura 52 - Obter ler e fazer mapeamento do Proof-Of-Delivery [referência própria]

Dentro do mesmo ciclo vamos:

1. Gerar uma *string JSON* com o output do *Mapper* (*RenderJSON*, figura 53);
2. Fazer uma chamada *RESTAPI* onde vamos dar upload do *output* do *RenderJSON* (*InvokeRESTAPI*, figura 53);
3. E por último eliminar o ficheiro do *FTP* (*FTPDeleteFile*, figura 53);

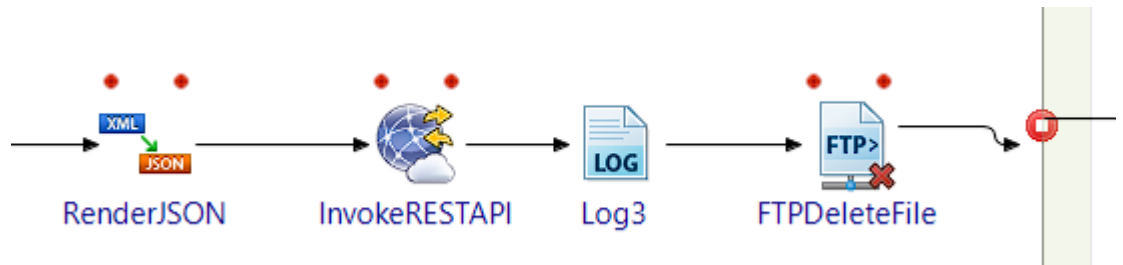


Figura 53 - Criar string Json, chamada RestAPI e eliminar ficheiro [referência própria]

4. Conclusão e trabalhos futuros

No início houve bastantes dificuldades com a integração neste estágio. Este projeto foi de elevada dificuldade, pois foi necessário estudar múltiplas áreas do conhecimento e ao contrário do percurso académico, num cenário de trabalho a ajuda é muito menor, logo o estagiário tem de ser muito mais independente.

Isto fez com que o primeiro mês, fosse o mês mais complicado de estágio. O autor foi instruído apenas para estudar o que tinha sido feito previamente pela empresa. Depois desse mês foi inicializado o primeiro projeto que foi referenciado no capítulo 3.5.1. Este projeto fez imensa diferença, visto que foi notado que ao trabalhar com as ferramentas aprendidas ao longo do estágio, a aprendizagem foi mais rápida do que apenas estudando.

Ao longo deste estágio foram aprendidas bastantes ferramentas que o autor foi usando e assim ganhando novas competências que irão ser muito importantes no futuro, não só em termos de programação, mas também ser mais autónomo e facilitando a resolução de alguns problemas.

Algo que na opinião do autor foi a parte mais importante deste estágio foi a integração numa equipa. Com isso foi ganha a experiência em saber como funciona as dinâmicas de trabalho de uma equipa, de como interagir com colegas de trabalho e de como funciona a hierarquia numa equipa. Também foi aprendido bastante com os colegas de trabalho da Rangel não só em termos das competências científicas, mas também como agir numa empresa e numa equipa.

Ao longo deste estágio, com a duração de 9 meses, foram apresentados 22 casos, conseguindo concluir os 22. A maioria dos casos foram resolvidos com facilidade, exceto o primeiro caso e o caso da Alcotrans, ambos referidos no capítulo 3.5.

Por fim, diria que este estágio foi um sucesso, devido às competências que o autor desenvolveu enquanto participou no estágio e devido ao interesse da Rangel, no seguimento do projeto com a empresa depois do final do estágio.

5. Bibliografia

- [1] Solutions R. História da Rangel [Internet]. Rangel Logistics Solutions. 2022 [cited 20 January 2022]. Available from: <https://www.rangel.com/pt/a-rangel/historia/>
- [2] Solutions R. Quem Somos [Internet]. Rangel Logistics Solutions. 2022 [cited 20 January 2022]. Available from: <https://www.rangel.com/pt/a-rangel/sobre-nos/quem-somos/>
- [3] Imagem retirada de <https://www.rangel.com/pt/a-rangel/sobre-nos/quem-somos/>
- [4] The Portal Company. 2022. What are Portals? - The Portal Company. [online] Available at: [<https://www.theportalcompany.com/what-are-portals/>](https://www.theportalcompany.com/what-are-portals/) [Accessed 21 January 2022].
- [5] Imagem retirada de <https://medium.com/@thshaunagraham/the-evolution-of-web-2-0-to-web-3-0-25218fcc33fd>
- [6] Imagem retirada de <http://what-when-how.com/portal-technologies-and-applications/evolution-of-portals/>
- [7] Corporate Web Portal Solutions: Prerequisite of Data Consolidation [Internet]. softengi.com. 2022 [cited 23 January 2022]. Available from: <https://softengi.com/blog/corporate-web-portal-solution/>
- [8] Imagem retirada de <https://www.gulisons.com/web-development/>
- [9] Imagem retirada de <https://marquesfernandes.com/es/tecnologia-es/que-es-un-frontend-desarrollador-y-que-hace/>
- [10] O que é a arquitetura da aplicação? [Internet]. www.redhat.com. 2022 [cited 22 January 2022]. Available from: <https://www.redhat.com/pt-br/topics/cloud-native-apps/what-is-an-application-architecture>
- [11] Imagem retirada de Novickis, Leonids & Vinichenko, Svetlana & Sotnichoks, Mihails & Lesovskis, Andrejs & Amalitskaya, Darja. (2015). Graph Models and GeoData Based Web Portal in Cargo Transportation. Applied Computer Systems. 17. 10.1515/acss-2015-0005.
- [12] Lyytinen K, Loucopoulos P, Mylopoulos J, Robinson B. Design Requirements Engineering: A Ten-Year Perspective. Berlin, Heidelberg: Springer Berlin Heidelberg; 2009.
- [13] Millett S, Tune N. Professional domain-driven design patterns. Indianapolis: Wrox; 2015.

[14] Stack Overflow Developer Survey 2021 [Internet]. Stack Overflow. 2022 [cited 20 June 2022]. Available from: <https://insights.stackoverflow.com/survey/2021#section-most-popular-technologies-integrated-development-environment>

[15] Programming Language | What is Programming Language - Javatpoint [Internet]. www.javatpoint.com. 2022 [cited 23 June 2022]. Available from: <https://www.javatpoint.com/programming-language>

[16] The A-Z of Programming Languages: C# [Internet]. Computerworld. 2022 [cited 23 May 2022]. Available from: https://www2.computerworld.com.au/article/261958/a-z_programming_languages_c_/?

[17] Chamberlin D. Early history of SQL. IEEE; 2012.

[18] Usage Statistics of JavaScript as Client-side Programming Language on Websites, June 2022 [Internet]. W3techs.com. 2022 [cited 23 May 2022]. Available from: <https://w3techs.com/technologies/details/cp-javascript>

[19] Learn to style HTML using CSS - Learn web development | MDN [Internet]. Developer.mozilla.org. 2022 [cited 23 May 2022]. Available from: <https://developer.mozilla.org/en-US/docs/Learn/CSS>

[20] Why HTML is Not a Programming Language by Ben Romy - iSchool | Syracuse University [Internet]. iSchool | Syracuse University. 2022 [cited 23 May 2022]. Available from: <https://ischool.syr.edu/why-html-is-not-a-programming-language/>

[21] HTML: Linguagem de Marcação de Hipertexto | MDN [Internet]. Developer.mozilla.org. 2022 [cited 23 May 2022]. Available from: <https://developer.mozilla.org/pt-BR/docs/Web/HTML>

[22] TypeScript Programming with Visual Studio Code [Internet]. Code.visualstudio.com. 2022 [cited 23 May 2022]. Available from: <https://code.visualstudio.com/docs/languages/typescript>

[23] SQL Server Management Studio (SSMS): The Definitive Guide (2022) [Internet]. Nill2Bill. 2022 [cited 23 May 2022]. Available from: <https://nill2bill.com/sql-server-management-studio-ssms/>

[24] 2007 Community Choice Awards - SourceForge Community Blog [Internet]. SourceForge Community Blog. 2022 [cited 23 May 2022]. Available from: <https://sourceforge.net/blog/cca07/>

[25] Imagem retirada de <http://myrangel.com>